MARINHA DO BRASIL
DIRETORIA DE ENSINO DA MARINHA
CENTRO DE INSTRUÇÃO ALMIRANTE ALEXANDRINO

CURSO DE APERFEIÇOAMENTO AVANÇADO EM
SISTEMAS DE ARMAS

TRABALHO DE CONCLUSÃO DE CURSO

MÉTODOS DE PROTEÇÃO CIBERNÉTICA: da concepção ao desenvolvimento do software de sistemas de combate

1º Ten GIOLIANO DE OLIVEIRA BRAGA

Rio de Janeiro
2023

1º Ten GIOLIANO DE OLIVEIRA BRAGA

MÉTODOS DE PROTEÇÃO CIBERNÉTICA: da concepção ao desenvolvimento do software de sistemas de combate

Monografia apresentada ao Centro de Instrução Almirante Alexandrino como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado em Sistema de Armas.

Orientador:
CC (EN) Vinícius Figueiredo Figueiredo dos Santos

Coorientadores:
Eng. José Fernando Maria Bianco Filho
CF (RM1) Wagner Santana de Freitas

CIAA
Rio de Janeiro
2023

Braga, Gioliano de Oliveira.
MÉTODOS DE PROTEÇÃO CIBERNÉTICA: da concepção ao desenvolvimento do software de sistemas de combate / Gioliano de Oliveira Braga. Rio de Janeiro, 2023.
57 f.: 21 x 29,7 cm
Orientador: CC (EN) Vinícius Figueiredo dos Santos.
Trabalho de Conclusão de Curso – Centro de Instrução Almirante Alexandrino, Curso de Aperfeiçoamento em Sistemas de Armas, 2023.

1. Segurança Cibernértica. 2. Sistema de Combate. 3 Sistemas Navais. I. Braga, Gioliano de Oliveira. II. Centro de Instrução Almirante Alexandrino. III. MÉTODOS DE PROTEÇÃO CIBERNÉTICA: da concepção ao desenvolvimento do software de sistemas de combate.

1º Ten GIOLIANO DE OLIVEIRA BRAGA

MÉTODOS DE PROTEÇÃO CIBERNÉTICA: da concepção ao desenvolvimento do software de sistemas de combate

Monografia apresentada ao Centro de Instrução Almirante Alexandrino como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado em Sistema de Armas.

Aprovada em _____

Banca Examinadora:

CF (RM1) Wagner Santana de Freitas – CIAA _____

CC (EN) Vinícius Figueiredo dos Santos, M.Sc. – DSAM _____

Eng. José Fernando Maria Bianco Filho – PUC-Rio _____

CT Françoa Taffarel Rosário Corrêa – CCEMSP _____

CIAA
Rio de Janeiro
2023

AGRADECIMENTOS

A Deus, pelo dom da vida, e por ajudar a superar os obstáculos nela existentes. Aos meus pais, que sempre priorizaram a educação mesmo em situações de grande dificuldade, mas mantiveram sua fé. À minha esposa Simone que me acompanhou nessa enfadonha jornada acadêmica e por tolerar os momentos em que estava presente em corpo, mas com a mente atarefada com os estudos. Ao meu orientador CC (EN) Vinícius Figueiredo pelo apoio incansável na pesquisa documental e correções ao longo deste trabalho que permitiram a rica confecção técnica desta obra e ao seu colega CC (T) Alisson por transmitirem seus conhecimentos. Aos coorientadores, engenheiro José Bianco e CF Wagner pelos ensinamentos em sala de aula que serviram como base sólida na elaboração deste trabalho acadêmico. Ao CMG (RM1) Asch por indicar os pontos em que a minha atenção deveria estar voltada para concentrar-me nos problemas a serem resolvidos e ao meu amigo CT Taffarel por indicar o rumo a ser navegado dentro dos assuntos de cunho cibernético.

“Sempre que vais atacar e combater, deves conhecer primeiro os talentos dos servidores do inimigo, e assim podes enfrentá-los segundo suas capacidades.”

Sun Tzu

MÉTODOS DE PROTEÇÃO CIBERNÉTICA: da concepção ao desenvolvimento do software de sistema de combate

Resumo

O exponencial avanço da dependência tecnológica e o crescimento da quantidade de sistemas computacionais no setor marítimo constituem um campo complexo que envolve a integração de sistemas de Tecnologia da Informação e Tecnologia Operacional. Ciberataques à Sistemas de Infraestrutura Críticas podem causar danos severos aos seres humanos e por isso que tais sistemas devem ser protegidos contra essas intrusões. Sendo assim, este trabalho apresentará uma proposta de teste de segurança cibernética para Sistemas Navais, baseado nas características do complexo software do sistema de combate, a fim de eliminar diversas vulnerabilidades que são as principais portas de entrada de ataques hackers. Baseados em estudos científicos anteriores, tal recomendação de teste irá expor técnicas, métodos, ferramentas e uma estrutura de orientação básica de segurança para serem aplicados em empresas e organizações governamentais a fim de aumentarem a cibersegurança, em especial contra os novos ciberataques usando ransomwares. As técnicas e ferramentas utilizadas para realizar o teste de segurança foram expostas e divididas em etapas ao longo do modelo em V que representa o ciclo de vida de desenvolvimento de sistema para tornar prático e sistemático o teste proposto. De maneira igual, o trabalho em questão visa contribuir para a sedimentação da mentalidade cibernética na Marinha do Brasil, bem como a capacidade de proteção de seus sistemas navais, proporcionando uma maior proteção cibernética aos softwares de sistemas críticos nacionais.

Palavras-chave: Sistemas navais. Segurança cibernética. Sistema de combate. Ransomware.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AIS	*Automatic Identification System*
BIOS	*Basic Input/Output System*
CASNAV	Centro de Análises de Sistemas Navais
CASA	*Common Architecture System Assurance*
CASOP	Centro de Apoio a Sistemas Operativos
CAv	Controle de Avarias
COTS	*Commercial off-the-shelf*
CGI	*Common Gateway Interface*
C4I	*Command, Control, Communications, Computers and Intelligence*
DCTIM	Diretoria de Comunicações e Tecnologia da Informação da Marinha
DNS	*Domain Name System*
DT&E	*Development Tests and Evaluation*
EUA	Estados Unidos da América
GPS	*Global Positioning System*
HTTP	*HyperText Transfer Protocol*
IEEE	*Institute of Electrical and Electronics Engineers*
IC	Infraestruturas Críticas
ICS	*Industrial System Control*
IPqM	Instituto de Pesquisas da Marinha
MB	Marinha do Brasil
NIST	*National Institute of Standards and Technology*
OT&E	*Operational Tests and Evaluation*
P&D	Pesquisa e Desenvolvimento
SDO	Sistemas Digitais Operativos

SSL/TLS	*Secure Sockets Layer / Transport Layer Security*
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicações
TO	Tecnologia Operacional
USB	*Universal Serial Bus*
USS	*United States Ship*
US Navy	*United States Navy*
VPN	*Virtual Private Network*

SUMÁRIO

1 INTRODUÇÃO

O aumento da dependência tecnológica e o crescimento da quantidade de sistemas ciberfísicos[1] no setor marítimo constituem um campo complexo. Uma questão quanto a segurança desses sistemas deve ser estudada: como os dados armazenados e o acesso aos sistemas serão protegidos diante de ameaças cibernéticas?

Hoje em dia, é notório os ataques às cadeias de suprimento de informações, que diferente da cadeia de suprimento logística, são sistemas que suprem demais sistemas ou um sistema central com dados e informações, no qual todos os processos são geridos por um software que integra vários setores tecnológicos [1]. Essas violações pretendem alcançar o máximo possível de vítimas ou causar danos a esses setores de modo a debilitá-los. Tais ataques tem diversos alvos, inclusive os sistemas marítimos internacionais, o que mostra o quanto ainda estamos vulneráveis no espaço cibernético por falhas de segurança de software e hardware [2].

Um exemplo claro disso é uma das maiores violações de segurança cibernética[2] do século XXI, ocorrida no ano de 2022, chamado de “Hack ao SolarWinds”, envolvendo o software Orion, uma ferramenta de monitoramento, análise e gerenciamento de plataformas de Tecnologia da Informação (TI), conectadas em uma rede de servidores de escala global, da empresa SolarWinds [3]. Essa intrusão afetou mais de 30 mil organizações públicas e privadas de todo mundo, devido ao software possuir acesso privilegiado aos sistemas de TI para obter dados de registro de desempenho dos sistemas das empresas [3].

Um pacote de atualização do software Orion distribuiu o arquivo infectado às vítimas do ataque, que tiveram seus arquivos criptografados. Essa ameaça ficou

[1]Sistemas ciberfísicos - são compostos por elementos computacionais em estreita relação com o ambiente físico, com o intuito de monitorar e controlar entidades e processos físicos em tempo real a partir do ambiente virtual [4].

[2]Segurança cibernética - busca proteger aquilo que é vulnerável na Tecnologia da Informação e Comunicações (TIC): informação digital e objetos não-informacionais, como eletroeletrônicos ou dispositivos inteligentes, por exemplo. A segurança cibernética também se preocupa com a proteção de sistemas de controle (hardware e software) e de Infraestruturas Críticas (IC), incluindo, assim, tudo no domínio cibernético [5].

conhecida como *ransomware*[3] "*Sunburst*" e se misturou com as atividades legítimas do Orion, tornando-o indetectável mesmo por software antivírus [3].

O ataque *ransomware*, como o causado pelo *NotPetya*, ocasionou em 2017, atrasos e interrupções em roteiros globais do transporte marítimo da empresa Moller-Maersk por semanas e causou perdas de centenas de milhões de dólares. O evento começou com a infecção de uma estação de trabalho de um único usuário e levou apenas sete minutos para se propagar amplamente e paralisar os serviços de TI da empresa [6].

O *Ryuk*, outro ataque *ransomware* ocorrido em 2019, foi observado pela Guarda Costeira dos Estados Unidos e atacou uma instalação através de e-mails que cativaram usuários a acessar links infectados (técnica conhecida como *phishing*[4]). O *Ryuk* teve um impacto significativo em TI, que também se propagou para o sistema de controle industrial (ICS, do inglês *Industrial Control System*) usado para monitoramento de instalações e movimentação de cargas [6].

A empresa Allianz, multinacional alemã que trabalha principalmente com seguros e gestão de ativos emite, anualmente, um relatório de riscos às negociações em diversos setores da indústria e comércio. No seu diagnóstico emitido em 2023, relata que as maiores ameaças atuais são a interrupção das cadeias de suprimentos[5] mundiais, como o comércio marítimo por incidentes cibernéticos, que podem deixar inativos sistemas industriais críticos como o transporte marítimo [7]. Diante de tais ameaças o desenvolvimento de um sistema de proteção cibernética mais sofisticado para no setor marítimo é necessário [6].

O uso de tecnologias de software avançadas e complexas em um navio está mais frequente em virtude das necessidades de demanda globais e das novas tecnologias empregadas nas guerras atuais. Basicamente, os navios de guerra

[3]*Ransomware* - é um tipo de malware (código malicioso) que impede que os usuários acessem seus recursos de dispositivos de computação e/ou dados pessoais usando vários métodos. Essa ameaça deixa o computador da vítima funcional para exibir a nota de resgate (instruções de pagamento) para os dados no dispositivo da vítima (seja um computador, servidor, tablet, smartphone ou dispositivo de Internet das Coisas) tornam-se inutilizáveis até que um resgate para remover a restrição seja pago. O *ransomware* é um problema cada vez maior que atinge indivíduos, o setor público, multinacionais, pequenas e médias empresas (PMEs) e aumentam ao passar dos anos [8].

[4]*Phishing* - técnica que visa capturar dados dos usuários. Chega por meio de mensagens eletrônicas falando de temas que atraem a atenção dos usuários, para fazê-los acessar links maliciosos ou instalar *malware* [9].

[5]Cadeia de suprimentos – é um sistema de organização de processos, recursos, pessoas, atividades e informações que mantêm esforços para realizar o transporte de cargas e produtos [10].

possuem um conjunto de sistemas, chamado de Sistema Naval[6]. Esse sistema é constituído pelos Sistemas de Navegação, Sistema de Comunicação, Sistema de Gerenciamento Integrado da Plataforma, Sistema de Sensores e Sistema de Combate[7], conforme detalhado na Tabela 1.

Diante desse cenário, o Sistema de Combate, com um complexo software de compilação e fusão de dados, pode ser afetado, de forma similar, por ciberataques ou por intrusões ao sistema, o que pode envolver grandes riscos a vida humana. Tal sistema está integrado a diversos outros sistemas como Sistema de Identificação Automática (AIS, do inglês *Automatic Identification System*), Sistema de Posicionamento Global (GPS, do inglês *Global Positioning System*) e o Sistema de Rede de Comunicações, que são potencialmente vulneráveis do ponto de vista cibernético [6].

A integração desses diversos sistemas heterogêneos ao Sistema de Combate aumenta a sua vulnerabilidade. Este Sistema Naval possui elevada complexidade, ou seja, é um "Sistemas de Sistemas" (SoS, do inglês *System of Systems*), o que dificulta consideravelmente a análise de vulnerabilidades de segurança contra ataques cibernéticos [2] por haver interrelações com os demais sistemas e subsistemas.

Tabela 1 - Sistemas encontrados em um navio de guerra

Sistema	Composição
Navegação	Radares de navegação, sistema do ecobatímetro, odômetro, anemômetro e giroscópio, luzes de navegação, sineres, GPS, AIS, *Electronic Chart Display and Information System* (ECDIS, sigla do inglês).
Comunicação	Rádios HF, VHF e UHF, comunicação via satélite, Banda-X, Banda-Ku, link de dados, Sistema Global de Socorro e Segurança Marítima (GMDSS, sigla do inglês), rede local interna (LAN, sigla do inglês), sistema de TV interno.
Gerenciamento Integrado da Plataforma	Sensores do navio que informam a situação de equipamentos como máquinas elétricas, motores diesel, máquinas hidráulicas e de sistemas dedicados como sistema de gerenciamento da propulsão e energia (PMS, do inglês *Power Management System*), sistema de níveis de tanques de água e combustível, sistema de distribuição de energia, sistema de refrigeração, sistema de osmose reversa, sistema de combate ao incêndio e alagamento, ecobatímetro, odômetro, anemômetro e giroscópio.

[6]Sistema Naval – são sistemas que abrangem uma ampla gama de tecnologias e subsistemas essenciais para a operação de embarcações navais no cumprimento de missões [11].

[7]Sistema de combate - tem a função de detectar, rastrear, identificar e (se necessário) engajar alvos dentro do alcance de seus Sistema de Sensores e Sistema de Armas [12].

Sensores	Radares de superfície, radares aéreos, radares de busca combinada, sonar, alças eletro-ópticas, AIS, *Identification Friend ou Foe* (IFF, sigla do inglês).
Combate	Sistemas táticos de superfície, aéreo, submarino, guerra eletrônica, sistema de armas de superfície, aéreo, submarino e eletrônico, além da integração dos sistemas de navegação e controle de avarias (CAv), comunicação externa, sistema de controle de voo, sistema de C4I e sistema sensores que alimentam o sistema de combate com informações relevantes ao cenário tático.

Fonte: [13] e [14]

O Sistema Naval tem numerosas vulnerabilidades cibernéticas semelhantes aos outros tipos de sistemas de informação. Pois os controles de segurança implementados contra vulnerabilidade cibernética não são suficientes, resultando em uma fragilidade perante os ataques cibernéticos [15]. Surge um novo desafio, problemas de segurança em tais sistemas podem comprometer seu desempenho e a sua função crítica de combate, o que pode atrapalhar a conclusão de missões navais em quaisquer circunstâncias e causar acidentes ou até perdas de vidas [16].

Explorar técnicas e métodos para desenvolver um teste de segurança para Sistemas Navais é necessário para garantir a proteção desses sistemas. Além disso, é importante evidenciar as boas práticas que devem ser tomadas no que diz respeito a segurança cibernética, especialmente as ameaças *ransomware* que estão atualmente causando danos a diversas organizações pelo mundo.

Os ataques cibernéticos aos meios navais ocorrem devido a vulnerabilidades encontradas nos diversos sistemas marítimos. Desta forma, as soluções técnicas de proteção cibernética devem ser estudadas para tornar tais sistemas robustos a essas ameaças. Nesse trabalho será discutido em que momento e de que forma os testes podem ser feitos ao longo do ciclo de vida de um projeto de sistema naval embarcado.

Conhecer os inúmeros ciberataques é desafiador, porém eles devem ser estudados por completo para se desenvolver formas de defesa mais efetivas.

1.1 Motivação

Alguns programas de desenvolvimento militar da Marinha do Brasil (MB) como o Programa de Desenvolvimento de Submarino (PROSUB), Programa Fragatas Classe “Tamandaré” (PFCT), Programa Antártico Brasileiro (PROANTAR), Sistema de Gerenciamento da Amazônia Azul (SisGAAz), Projeto MANSUP (Míssil Antinavio – Superfície), entre outros, deverão possuir desde a sua concepção, além da preocupação com a proteção dos dados digitais, métodos de segurança capazes de fornecer a devida blindagem cibernética durante todo o ciclo de vida. Tais projetos são Infraestruturas Críticas[8] (IC) Navais que devem ser protegidas por incentivos a pesquisas [5], para estimular a indústria de Tecnologia de Defesa, em particular à naval, aumentando como um todo a capacidade cibernética da MB. A preocupação em aprimorar as defesas cibernéticas deverá estar presente desde o início do ciclo de vida dos sistemas, de modo a ter maior segurança, mais eficiência e economia evitando falha de projetos e retrabalho após o sistema estar desenvolvido.

1.2 Objetivos

Este trabalho contribuirá com a proteção de Organizações Militares da MB contra agentes adversos no campo cibernético para no mínimo, assegurar a cibersegurança de IC navais ou que sejam responsáveis pela gestão de conhecimento sensível [17].

Nesta pesquisa, o sistema de combate do navio de guerra da Marinha será considerado como IC. Os demais sistemas navais serão tratados como cadeia de suprimento de informações que transmitem dados e informações para o sistema de

[8]Infraestruturas Críticas (IC) Navais - infraestruturas organizacionais que possuem redes cabeadas e sem fio, como equipamentos criptográficos, sistemas de armas, sistemas de combate, equipamentos embarcados em meios operativos, dentre outros [5] que possuem dimensão estratégica, uma vez que desempenham papel essencial tanto para a segurança e soberania nacionais [18].

combate que irá gerenciá-los a fim de apoiar a decisão do comando no cumprimento de missões.

Este trabalho acadêmico servirá de base para estudos de Pesquisa e Desenvolvimento (P&D), elaboração de normas e documentações técnicas na área de segurança cibernética, tendo em vista, a latente ameaça dos ataques cibernéticos, bem como fomentar a mentalidade de segurança cibernética em desenvolvedores e operadores de sistemas, em especial àqueles relacionados com atividades marítimas.

1.3 Metodologia

Inicialmente deverá ser entendida a atividade de alguns ataques como *malwares*, foça bruta, injeção *phishing*, *ransomware*, às falhas de aplicações web, falsificação de identidade e outros, para então, conhecer os pontos de vulnerabilidade em potencial dos sistemas navais [6]. Desta forma, no capítulo 2 serão descritos, alguns ataques de *malwares* às vulnerabilidades normalmente encontradas em sistemas, embarcações e estruturas marítimas, bem como os danos e impactos causados e uma relação quantitativa de ataques *ransomware* ao setor marítimo.

Serão descritas também dicas básicas de boas práticas para proteção contra novos ataques, como *ransomware*, para que tais ameaças não se instalem principalmente no ambiente de desenvolvimento de software. Será explicitada as boas práticas para prevenção de incidentes através da apresentação de uma estrutura básica de segurança cibernética baseado em guias do Instituto Nacional de Padrões e Tecnologia (NIST, do inglês *National Institute of Standards and Technology*) dos Estado Unidos da América (EUA).

Os detalhes das particularidades do sistema de combate e de seu software serão apresentados no capítulo 3.

No capítulo 4 apresentará métodos e técnicas disponíveis em literaturas especializadas em proteção cibernética e utilizados pela *United States Navy* (*US Navy*), além de, estratégias para realização dos testes de segurança aprimorados pelos autores Cheol-Gyu Yi e Young-Gab Kim, na obra [16], com aplicação em

Sistemas Navais Embarcados, como os sistemas de combate presentes nos navios da Marinha do Brasil através de abordagem qualitativa e pelo método hipotético-dedutivo.

Dessa forma, será elaborada uma solução técnica para eliminar ou mitigar vulnerabilidades de sistema embarcados de navios militares através da avaliação do teste de segurança proposto, conforme a fase de desenvolvimento do ciclo de vida de sistemas de software de defesa [19] apresentadas pelo modelo em V.

Finalmente, será apresentado as etapas de execução do teste de segurança baseado no modelo em V apresentado, desde a concepção do projeto de software até a sua operação, além de ferramentas de segurança que podem ser utilizados em plataformas de sistema operacional Linux e as suas utilidades.

Na elaboração deste trabalho normas vigentes da Marinha do Brasil foram consultadas, o EMA-419 referente à Doutrina Cibernética da Marinha e o PEM-2040 tratando do Plano Estratégico da Marinha, o Decreto nº 10.569, de 9 de dezembro de 2020 que aprovou a Estratégia Nacional de Segurança de Infraestruturas Críticas, o MD-40-M-1 um Manual de Boas Práticas para a Gestão do Ciclo de Vida de Sistemas de Defesa, o Guia de Aperfeiçoamento da Segurança Cibernética para Infraestrutura Crítica, a norma de Gestão de risco de *ransomware* do *National Institute of Standards and Technology* e o livro de Nihad A. Hassan fornecem uma estrutura de segurança de dados e um guia de proteção e recuperação contra ataques *ransomware*, além de diversos artigos encontrados na plataforma do *Institute of Electrical and Electronics Engineers* (IEEE, sigla do inglês) e o Guia para segurança para o desenvolvimento de software emitido pelo Ministério do Interior e Segurança da Coréia do Sul que versam sobre técnicas e métodos de proteção cibernética que são aplicados a sistemas diversos, inclusive no sistema de combate.

Tais literaturas permitiram a realização de uma pesquisa básica estratégica e descritiva para que a solução deste trabalho sirva como base científica para outros com problemas semelhantes. A abordagem qualitativa empregada durante a análise de todo o procedimento bibliográfico e documental utilizado, permitirá atingir o objetivo geral do trabalho.

1.3.1 Quanto aos meios

As fontes do trabalho foram publicações e normas vigentes da Marinha do Brasil como o EMA-419 e o PEM-2040. Documentos de Institutos Internacionais renomados, artigos de jornais digitais da internet, em especial do *Institute of Electrical and Electronics Engineers*, além de livros relacionados ao tema e livros referenciados em aulas ministradas durante o Curso de Aperfeiçoamento em Sistema de Armas.

Com a abordagem qualitativa foi possível utilizar o método hipotético-dedutivo para alcançar uma solução ao problema encontrado: os sistemas navais embarcados, assim como o sistema de combate, não são robustos a ataques cibernéticos. A partir dessa hipótese será deduzido métodos de proteção cibernética para encontrar uma solução.

1.3.2 Limitações do método

Uma das principais limitações e dificuldades encontradas para o desenvolvimento desse trabalho é a sua aplicação prática, pois é admissível que há deficiência no setor de segurança cibernética, como mostrado no Índice de Potência Cibernética das Nações em [20]. Principalmente em grau técnico, há pouca informação (publicações, normas técnicas entre outros documentos) disponível no Brasil referente a técnicas e métodos de proteção cibernética, o que levou esta pesquisa acadêmica a exploração de muitos trabalhos estrangeiros para que a qualidade desta obra fosse a mais próxima possível do que é utilizado atualmente.

Na Marinha do Brasil não existem normas e publicações que abordam procedimentos, técnicas ou ferramentas de software que pudessem ser utilizados para esta pesquisa.

O ponto de vista prático, o método desenvolvido neste trabalho possui uma limitação que será explorada na seção 4.4

1.3.3 Universo e Amostragem

Por meio de artigos ostensivos sobre métodos e técnicas de proteção cibernética, o trabalho utilizou os sistemas de combate dos navios de guerra e seu software como objeto de estudo, ampliando-o para Sistemas Navais Embarcados por apresentarem características semelhantes.

Nesta pesquisa os sistemas que suprem o sistema de combate com dados e informações serão considerados como cadeia de suprimento de informações. Os sistemas constituintes dessa cadeia podem ser considerados como Sistema Naval.

Assim, os sistemas navais são considerados Sistemas Digitais Operativos[9] (SDO) pois são empregados em operações táticas bem como os seus subsistemas.

1.4 Contribuição

No que se diz respeito ao desenvolvimento de projetos de sistemas robustos à ataques cibernéticos, os testes de segurança para software dos Sistemas Navais Embarcados têm grande capacidade de melhorar a cibersegurança dos meios da MB, frente às novas ameaças impostas pela tecnologia.

As principais contribuições desse trabalho serão:

1. Demostrar que embarcações e estruturas marítimas não estão ilesos à ataques cibernéticos e a classificação de ameaças nesse setor, bem como, uma estrutura de segurança cibernética para ser utilizado em ambientes de trabalho que possuam sistemas informatizados;

2. Estabelecimento de uma estrutura de teste de segurança de software com base nas características do software utilizado no sistema de combate;

[9]Sistemas Digitais Operativos: sistema computacional não administrativo de emprego tático, de controle da propulsão, de avarias e seus subsistemas associados, existentes nos meios operativos e nos simuladores do ambiente operativo de bordo. Compostos, obrigatoriamente, de componentes de hardware (computadores digitais e periféricos) e de componentes de software (programas controladores da operação), juntos com os sistemas analógicos da MB [21].

3. Detalhar o teste de segurança em cada uma das camadas dos Sistemas Navais Embarcados durante sua fase de concepção;

4. Reflexão acerca dos requisitos de técnicas e métodos de segurança para o projeto do software do sistema de combate durante o seu processo de desenvolvimento;

5. Desenvolvimento de teste de segurança a fim de remover antecipadamente vulnerabilidades do programa; e

6. Descrever um conjunto de ferramentas open-source[10] para sistemas operacionais Linux para testes de software.

2 VULNERABILIDADES E AMEAÇAS CONTRA SISTEMAS

Neste capítulo serão descritas as principais ameaças aos sistemas marítimos para facilitar o entendimento das vulnerabilidades e uma classificação das ameaças às embarcações e estruturas marítimas. Além disso, serão evidenciados os riscos do *ransomware* para o setor marítimo e alguns métodos que empresas e órgãos governamentais possam utilizar em sua estrutura de segurança da informação, adicionando boas práticas e mentalidade de cibersegurança para mitigar ataques *ransomware* contra seus sistemas.

2.1 Principais vulnerabilidades e ameaças

A Tabela 2 abaixo detalha uma exploração de ataques e pontos de entradas mais utilizados por intrusos para acessar recursos indevidos dos diversos sistemas de software. Este detalhe será aqui realizado para entender como os ataques são

[10]*Open Source* - o software “open source” é lançado por meio de um tipo de licença específica que disponibiliza legalmente o código-fonte aos usuários finais [23].

executados e a maneira com que administradores podem proteger adequadamente seus sistemas contra tais ameaças [22].

Tabela 2 – Ataques comuns as vulnerabilidades

TIPO DE EXPLORAÇÃO	DESCRIÇÃO	NOTAS
Senhas padrão ou nulas	Deixar as senhas administrativas em branco ou usar uma senha padrão definida pelo fornecedor do produto.	Geralmente associado a hardware de rede, como roteadores, firewalls, Virtual Private Networks (VPN, sigla do inglês) e dispositivos de armazenamento conectado à rede. Às vezes, os administradores criam contas de usuários privilegiados rapidamente e deixam a senha nula, criando um ponto de entrada perfeito para usuários mal-intencionados que descobrem a conta.
Compartilha-mento de chaves padrão	Às vezes, os serviços seguros empacotam chaves de segurança padrão para fins de desenvolvimento ou testes de avaliação. Se essas chaves permanecerem inalteradas e forem colocadas em um ambiente de produção na Internet, todos que possuírem as chaves terão acesso a qualquer informação confidencial.	É mais comumente encontrado em pontos de acesso de redes e dispositivos de servidores seguros pré-configurados.
Falsificação de IP	Uma máquina remota atua como um nó em sua rede local, encontra vulnerabilidades em seus servidores e instala um programa *backdoor*, como um *ransomware*, ou cavalo de Tróia para obter controle sobre os recursos de sua rede.	Dependo do sistema de destino serviços podem ser executados (como rsh, telnet, FTP e outros) que usam técnicas de autenticação baseadas na fonte, que não são recomendadas quando comparadas ao *Public Key Infrastructure* (PKI, sigla do inglês) ou outras formas de autenticação criptografada usada em ssh ou SSL/TLS.
Escuta	Coleta de dados, por um terceiro, que passam entre dois nós ativos em uma rede	As medidas preventivas incluem serviços com troca de chaves criptográficas, senhas de uso único ou autenticação criptografada para evitar espionagem de senhas; criptografia forte durante a transmissão também é recomendada.
Vulnerabilida-des de serviços	Um invasor encontra uma falha ou brecha em um serviço executado na rede; através desta vulnerabilidade, o invasor compromete todo o sistema e quaisquer dados que ele possa conter, podendo comprometer outros sistemas na rede.	Às vezes, os serviços podem ter vulnerabilidades que passam despercebidas durante o desenvolvimento e os testes; essas vulnerabilidades (como buffer overflows, onde invasores travam um serviço usando valores arbitrários que preenchem o buffer de memória de um aplicativo) podem fornecer controle administrativo completo a um invasor. Os administradores devem certificar-se de

		que os serviços não sejam executados como usuário root e devem ficar atentos a patches e atualizações de erratas para aplicativos de fornecedores ou organizações de segurança, como o Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança (CERT).
Vulnerabilidades dos aplicativos	Os invasores encontram falhas em aplicativos de desktop e estações de trabalho (como clientes de e-mail) e executam códigos arbitrários, implantam cavalos de Tróia para comprometimento futuro ou travam sistemas usando *ransomware*. Poderá ocorrer exploração adicional se a estação de trabalho comprometida tiver privilégios administrativos no restante da rede.	É imperativo informar os indivíduos sobre os riscos que correm quando instalam software não autorizado ou abrem anexos de e-mail não solicitados. As proteções podem ser implementadas de forma que o software cliente de e-mail não abra ou execute anexos automaticamente. Adicionalmente, a atualização automática do software das estações de trabalho utilizando Red Hat Network; ou outros serviços de gerenciamento de sistema podem aliviar os encargos das implantações de segurança

Fonte: [22]

Outra recomendação, não contida na Tabela 2, simples, mas não menos importante, é a proteção do *firmware* do Sistema Básico de Entrada/Saída (BIOS[11], sigla do inglês). É recomendado que uma senha de acesso deverá ser configurada para evitar que um invasor realize alterações nas configurações, impedindo-o de configurá-la a partir de um pendrive, ou acessando o modo de recuperação, ou como usuário especial [22]. Isso impede a inicialização desprotegida do sistema, forçando o atacante a lidar com um *firmware* cifrado antes mesmo do *boot*[12] ser iniciado, o que garante a integridade do Sistema Operacional original do sistema.

Nesse ponto de vista, muitas vulnerabilidades foram detectadas em *firmwares*. Os autores do artigo “Caracterização das vulnerabilidades dos roteadores Wi-Fi no mercado brasileiro”, em sua pesquisa, observaram, em média, 1344

[11]*Basic Input/Output System* (BIOS) - é um *firmware* (software no dispositivo de hardware que executa tarefas básicas de entrada e saída e instruções para comunicação entre outros dispositivos) não volátil usado para executar a inicialização do hardware durante seu processo de inicialização e para fornecer serviços de tempo de execução para sistemas operacionais e programas [25].

[12]*Boot* – processo que ao inicializar, garante que o computador cumpra duas funções. A primeira é o autodiagnóstico, testa o hardware para comprovar o funcionamento de componentes como HD, placas e memória. Em seguida, garante o carregamento dos arquivos exigidos pelo sistema operacional encontrado na máquina (Linux, macOS ou Windows) [26].

vulnerabilidades de kernel[13] e 72 fragilidades nas aplicações desses dispositivos de conexão de redes sem fio [24].

Muitas dessas fraquezas estão relacionadas a usuário e senhas fracas ou padrões de fábrica, além de vulnerabilidades de *buffer overflow* e até *zero-day*[14] [24].

2.2 Classificação das ameaças

Algumas ameaças às embarcações e estruturas marítimas, inclusive militares são classificadas pelo Instituto Americano de Petróleo (API, sigla do inglês *American Petroleum Institute*) [28] de acordo com as localizações dos atores envolvidos nos incidentes cibernéticos. Elas são muitas vezes relacionadas apenas a ataques externos ao ambiente de uso do sistema, mas também, como poucos lembram, os ataques podem ser internos ou até colaborativas (internas e externas), conforme o detalhamento para os meios navais abaixo:

1) Uma ameaça interna pode ser um membro da tripulação de uma embarcação ou um operador portuário que permite involuntária ou intencionalmente a penetração das barreiras de segurança cibernética (neste caso, plataformas e ferramentas de TI) ao operar no domínio ciberfísico sem práticas adequadas de segurança cibernética. Desde o uso de um dispositivo USB infectado por vírus até a abertura de um e-mail não solicitado infectado por malware [28];

2) As ameaças externas podem ser uma série de criminosos cibernéticos comuns, hackers, ativistas cibernéticos ou até adversários estatais, ou terroristas que utilizam métodos sofisticados para manipular, degradar ou assumir o controle de sistemas de TI e TO [28]; e

3) As ameaças colaborativas são uma combinação de atores de ameaças internas atuando com orientação de fontes externas [28].

[13]Kernel (ou núcleo) – componente central do sistema operacional responsável por gerenciar recursos de hardware como memória, dispositivos de entrada e saída e processador [27].

[14]Zero-Day - situação na qual há uma ameaça de computador capaz de explorar uma vulnerabilidade de segurança descoberta em sistemas computacionais e que não teve, ainda, correção disponibilizada pelo desenvolvedor ou fabricante [5].

2.3 Principais ataques cibernéticos a meios marítimos nos últimos anos

A proteção da cadeia de suprimento de informação, como o Sistema Naval, é importante. Por isso, foi elaborada a Tabela 3, baseada em acontecimentos reais, em que os ataques a estas cadeias causaram prejuízos e até fatalidades. Esses fatos estão detalhados na coluna “local e descrição”.

Além disso, observa-se que alguns ataques causaram a perda de vidas humanas durante a colisão de navios, fato preocupante, que torna cada vez mais fundamental a proteção cibernética robusta dos sistemas navais citados nessa pesquisa.

Tabela 3 – Ataques cibernéticos significativos a embarcações/estruturas marítimas

ÍNDICE	TIPO DE ATAQUE CIBERNÉTICO	ANO	LOCAL E DESCRIÇÃO
1.	***Ransomware* e *Phishing***	2021	*Hyundai Merchant Marine* (HMM, sigla do inglês), a principal transportadora marítima nacional da Coreia do Sul: recebe ataque cibernético resultando em acesso limitado ao sistema de e-mail Outlook.
2.	***Ransomware***	2020	Porto de Hormuz: a tentativa de ataque cibernético danificou alguns sistemas operacionais nos portos.
3.	*Malware*	2020	*Mediterranean Shipping Company* (MSC): por questões de segurança, os servidores da MSC foram encerrados para proteger os dados da empresa e, como resultado, o site da empresa foi retirado do ar. O ataque perturbou apenas processos de dados internos.
4.	***Ryuk Ransomware***	2019	O ataque teve como alvo um navio dos Estados Unidos da América (EUA), causando mineração crítica de credenciais. A Guarda Costeira e o Federal Bureau of Investigation (FBI, sigla do inglês) relataram que a falta de estratégias de segurança na embarcação foi o principal motivo do ataque. Foi constatado que todos os tripulantes da embarcação compartilhavam o mesmo login e senha do computador da embarcação. Além disso, o uso de dispositivos externos facilitou a tarefa do hacker. Outro erro crítico é a falta de antivírus.
5.	*Phishing*	2019	Os hackers obtiveram acesso não autorizado aos sistemas de computadores da James Fisher and Sons Pls (Reino Unido).
6.	***Ransomware***	2018	Hackers chineses atacaram empreiteiros da Marinha dos EUA.
7.	***Petya Ransomware***	2017	O ataque denominado Petya afetou servidores de computadores na Europa e na Índia. O malware criptografado foi direcionado a todos os serviços

			da empresa de navegação Maersk. Como resultado, 17 terminais de contentores foram afetados e mais de 200 milhões de dólares foram perdidos. O ataque destruiu gravemente o sistema operacional dos computadores ao infectar seu registro mestre de inicialização (MBR, sigla do inglês).
8.	GPS *spoofing*	2017	O ataque é relatado pela administração marítima dos EUA. O GPS de um navio no porto russo de Novorossiysk indicou uma localização errada. O ataque é provavelmente um teste de um novo sistema de falsificação de GPS.
9.	Ataque a sistemas de navegação	2017	O ataque ao sistema de navegação causou a colisão entre o United States Ship (USS, sigla do inglês) Fitzgerald e um navio porta-contêineres, causando a morte de 7 marinheiros. (na costa do Japão)
10.	Ataque a sistemas de navegação	2017	O ataque ao sistema de navegação causou a colisão entre um petroleiro e o USS John S. McCain perto da costa da Malásia: a morte de 10 marinheiros.
11.	Ataque a sistemas de navegação	2017	O ataque ao sistema de navegação causou a colisão entre o USS Lake Champlain e um navio pesqueiro sul-coreano.
12.	GPS *spoofing*	2013	Experiência realizada por uma equipe de pesquisa da Universidade do Texas para falsificar a posição de um iate.
13.	Vírus de computador dentro dos sistemas de controle	2012	Redes de comunicação instaladas na plataforma offshore de petróleo e gás no Golfo Pérsico foram atacadas
14.	*Phishing*	2010-2013	Os cibercriminosos desenvolveram uma entrada *backdoor* chamada "*Fucobha: the Icefog*" (japonês e sul-coreano)
15.	***Ransomware***	2011	As Companhias Navais da República Islâmica do Irã (IRISL, sigla do inglês) foram atacadas
16.	*Phishing*	2011-2013	Porto de Antuérpia, na Bélgica: Um grupo do crime organizado utilizou hackers sediados na Bélgica para controlar as redes informatizadas de empresas que operam no porto de Antuérpia.

Fonte: [12]

Na Tabela 3, pode-se verificar que os ataques à sistemas marítimos não são novidades e podem ocorrer tanto em estrutura de portos como em embarcações civis ou militares.

As ameaças de ataques *ransomware*, da Tabela 3, representaram 1/3 das ocorrências registradas entre 2010 e 2021. Estes ataques foram realizados sobre estruturas marítimas, o que é uma grande preocupação para o Brasil, pois 95% do comércio exterior ocorrem por vias marítimas [29]. Logo, a próxima seção irá detalhar este novo tipo de ataque.

2.4 Ransomware: um novo tipo de ataque

O *ransomware*, por definição, é um tipo código malicioso que impede que os usuários acessem seus dados pessoais, por meio de criptografia, usando vários métodos de ciberataques em conjunto. O computador da vítima permanece funcional apenas para pedir um resgate de pagamento para fornecimento da chave de criptografia dos dados pessoais da vítima [8]. O *ransomware SunBurst*, por exemplo, se misturou a atividades legítimas de antivírus e ficou por mais de 14 meses sem ser detectado, indicando que os ataques estão se aperfeiçoando cada vez mais com o passar do tempo [3].

Com o objetivo de evitar ou, ao menos, reduzir ataques *ransomware*, o NIST criou uma estrutura de segurança cibernética. O “Guia de aperfeiçoamento da segurança cibernética para infraestrutura crítica” [30], que orienta e considera as atividades e os riscos de segurança cibernética como parte dos processos de gerenciamento de riscos de uma organização.

Esse guia para toda organização se envolva no assunto de segurança cibernética, não somente os setores de TI ou TO, de forma que todos os setores se preocupem com a segurança cibernética da organização, pois todos podem ter seus sistemas invadidos ou violados [30].

A Tabela 4 contêm orientações básicas de boas práticas preventivas para que instituições se protejam de *ransomwares*. A tabela foi baseada na ideia de cinco funções de segurança cibernética: identificação, proteção, detecção, resposta e recuperação [30] e foi também baseada na estrutura de segurança da referência [31]. O objetivo aqui é fazer com que que as orientações da Tabela 4 sejam utilizadas por especialistas em segurança de TI, principalmente, durante a etapa de desenvolvimento de um projeto de software para sistemas navais, a fim de evitar as ameaças *ransomware*.

Tabela 4 – Orientações básicas de prevenção contra ameaças *ransomware*

1. Instruir os funcionários sobre como evitar infecções por ransomware. • Não abrir arquivos ou clicar em links de fontes desconhecidas; • Evitar sites e aplicativos pessoais; e • Não conectar dispositivos pessoais a redes de trabalho sem autorização prévia.

2. Evitar vulnerabilidades em sistemas que o ransomware possa explorar. • Manter os sistemas relevantes totalmente corrigidos**;** • Aplicar princípios de "zero trust" (modelo de segurança baseado na confiança zero, acesso aos ativos requer validação e autorização) em todos os sistemas em rede**;** • Permitir a instalação e execução apenas de aplicativos autorizados; e • Informar seus fornecedores de tecnologia sobre suas expectativas**.**
3. Detectar e interromper rapidamente ataques e infecções de ransomware. • Usar sempre software de detecção de malware, como software antivírus**;** • Monitorar continuamente os serviços de diretório; e • Bloquear o acesso a recursos da Web não confiáveis**.**
4. Dificultar a propagação do ransomware. • Usar contas de usuário padrão com autenticação multifator**;** • Introduzir atrasos de autenticação ou configurar o bloqueio automático de contas**;** • Atribuir e gerenciar a autorização de credenciais para todos os ativos e softwares da empresa; • Armazenar os dados em um formato imutável; e • Permitir o acesso externo a recursos de rede internos somente por meio de conexões VPN (rede virtual privada) seguras.
5. Facilitar a recuperação de informações armazenadas em um futuro evento de ransomware. • Fazer um plano de recuperação de incidentes**;** • Fazer backup de dados, proteger backups e testar a restauração**;** • Bloquear o acesso a recursos da Web não confiáveis; e • Manter uma lista atualizada de contatos internos e externos para ataques de ransomware, incluindo aplicação da lei, assessoria jurídica e recursos de resposta a incidentes.

Fonte: [31]

As boas práticas da Tabela 4 tem como objetivo evitar os pontos de vulnerabilidades apresentados na coluna "notas" da Tabela 2, estas vulnerabilidades, quando corrigidas, fornecem ao sistema uma boa proteção aos ataques cibernéticos.

3 DETALHES DO SISTEMA DE COMBATE

Possuir um software que seja estado da arte[15] para segurança cibernética dos sistemas de um navio de guerra é algo almejado por diversas Forças Navais, como citado na reportagem da *Thales Group* em [32], pois a utilização desse tipo de

[15]Estado da arte (SOTA, sigla do inglês *state-of-the-art*) – algo muito moderno que utiliza as ideias e métodos mais recente [33].

software ajudaria as marinhas de todo o mundo a implementar de forma mais fácil e econômica medidas de defesa cibernética para os seus navios [32].

A Marinha Norte Americana alcançou tal objetivo com o desenvolvimento de um sistema de proteção cibernético para os seus sistemas de combate dos navios de guerra como o sistema CASA[16] [34]. Pensando em ter objetivo similar para a Marinha do Brasil, este trabalho irá divulgar procedimentos que visam a proteção dos sistemas navais de embarcações militares, principalmente para o software do sistema de combate de um navio de guerra, utilizando ferramentas de detecção de vulnerabilidade.

3.1 Estrutura do software do sistema de combate

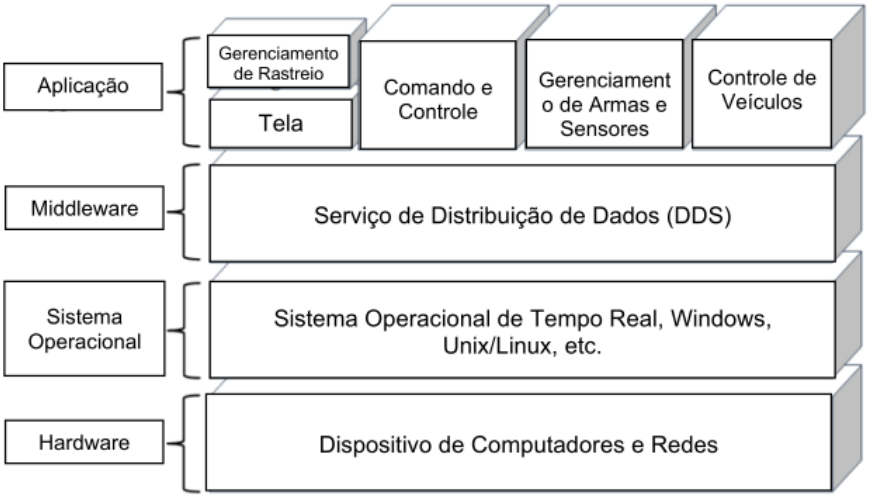
A estrutura base do software do sistema de combate é apresentado na Figura 1, que pode ser dividido nas seguintes camadas heterogêneas:

- Camada de aplicação: programas que ligam o usuário e o operador do sistema, por exemplo, o programa do sistema de combate (gerenciamento das armas e aplicação de comando e controle), o programa do gerenciamento dos sensores do navio, os programas de navegação, entre outros;
- Camada do *Middleware*: é basicamente uma interface de abstração entre o sistema operacional e a aplicação, que fornece comunicação e serviços com uma programação comum e padronizada;
- Camada do Sistema Operacional: é um software que atua como uma camada intermediária entre os componentes de hardware de um computador e os aplicativos de software. Ele desempenha um papel crucial na gestão e coordenação dos recursos do sistema, permitindo que os usuários interajam com o computador de maneira eficaz e os aplicativos funcionem corretamente; e

[16]*Common Architecture System Assurance* (CASA) - O software projetado para coletar, analisar e relatar todos os eventos relacionados à Segurança da Informação de um sistema, rede ou, potencialmente, de uma plataforma inteira, no caso o próprio sistema de combate. O seu projeto foi executado juntamente com o sistema de combate “Aegis” dos cruzadores e destroieres da US Navy [34].

- Camada do Hardware: parte física do sistema que possui uma programação interna específica para controle de cada componente conhecida como *firmware* (como por exemplo a BIOS).

Figura 1 - Estrutura do software do sistema de combate

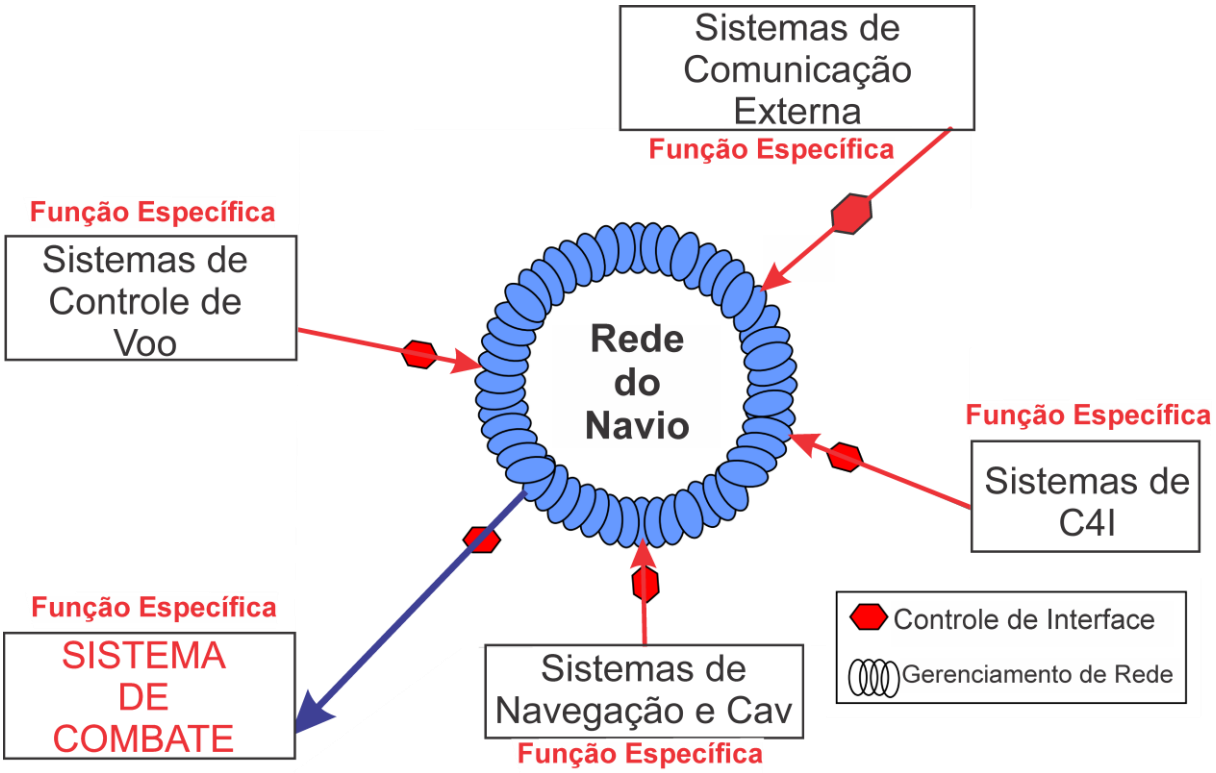


Fonte: [16]

O sistema de combate gerencia a parte lógica do sistema de armas de elevada complexidade operada por computadores integrados por meio de redes [12]. É composto por vários sistemas heterogêneos, que incluem sensores, armamentos, sistemas de navegação e outros, todos integrados em um navio de guerra. Esse é um sistema encarregado de gerar e compartilhar informações em situações táticas, conectando diferentes sistemas e conduzindo automaticamente tarefas de detecção, rastreamento de alvos, análise de ameaças, redirecionamento de armas, engajamento e avaliação de danos [16].

A Marinha dos Estados Unidos optou por empregar a abordagem modular de arquitetura aberta na criação de seus sistemas de combate [34]. Utilizou os princípios fundamentais de projeto modular, reutilizável, de melhor interoperabilidade e desempenho por meio de tecnologias avançadas [34]. O aplicativo CASA desempenha diversas funções críticas para missões, tais como gerenciamento de trajetória, comando e controle, o gerenciamento e engajamento de sensores e armas, em missões de combate. Pode facilmente adaptar-se às alterações de acordo com a missão designada e, em particular, às funções necessárias para um combate [14].

Figura 2 - Estrutura das funções em rede do navio de guerra



Fonte: próprio autor

Observa-se na Figura 2 que os sistemas de funções específicas: aviação, comunicação externa, C4I (sigla do inglês *Command, Control, Communications, Computers and Intelligence*) navegação e Controle de Avarias (CAv), são considerados cadeia de suprimento de informações, transportando dados táticos para o sistema de combate por meio da conexão de rede do navio [14], por isso a preocupação de uma melhor implementação da proteção cibernética do sistema de combate e também dos demais sistemas marítimos integrados.

Por razões econômicas e a necessidade de otimização de tempo, muitas organizações militares e industriais, pelo mundo, estão buscando *middlewares* em produtos comerciais prontos, COTS (sigla do inglês, *Commercial off-the-shelf*) [34] e [35], o que facilita o desenvolvimento de sistemas de código-fonte aberto.

Se o aplicativo do sistema de combate precisar ser alterado, as definições de configuração do sistema operacional também deverão ser atualizadas, pois os sistemas são altamente conectados entre si [16]. É necessário realizar um novo teste dos sistemas a fim de verificar se eles permanecem íntegros e não tenha surgido nenhuma vulnerabilidade adicional [16].

4 FASES, ETAPAS E METODOLOGIA DO TESTE DE SEGURANÇA

Infelizmente, é uma prática atual que os testes de segurança de softwares não sejam conduzidos sistematicamente durante todo o desenvolvimento dos diversos sistemas de software. Dessa forma, não há um sistema robusto a ataques cibernéticos em vulnerabilidades [16].

O problema associado aos testes de segurança é a falta de critérios específicos e sistemáticos definidos no desenvolvimento do software o que possibilitaria alta robustez desde a fase de produção do ciclo de vida do software [16].

No capítulo 4, será apresentada a definição de fases, etapas e uma metodologia para a realização do teste de segurança durante todo o ciclo de vida de desenvolvimento do software de sistemas navais. Na seção 4.1 será detalhada as fases e etapas do ciclo de vida para compreensão de quais momentos os testes devem ocorrer.

Em toda a seção 4.2 serão definidos critérios do teste de segurança bem como os requisitos necessários para a realização do teste, tais critérios e requisitos serão apresentados em tabelas.

O teste de segurança do software tem o intuito de identificar erros e falhas no aplicativo dos sistemas, eliminando de forma proativa as vulnerabilidades de segurança por meio de critérios que serão apresentados na seção 4.2.1.

Ao longo de toda a seção 4.3 as categorias de testes serão aprimoradas com base em [16] e divididas por camadas da estrutura do software do sistema de combate. Tal sistema foi escolhido por sua característica complexidade e heterogeneidade, o que permite um teste de segurança mais eficaz para corrigir as vulnerabilidades.

Para melhorar o teste de segurança, a seção 4.4 traz orientações básicas que sugerem ajustes.

A seção 4.5 apresentará uma visão dos processos de desenvolvimento e integração através do modelo em V aprimorado por [16].

Por fim, a seção 4.6 detalha os testes de segurança com o uso de ferramentas para o sistema operacional de código aberto Linux.

4.1 Fases e etapas do desenvolvimento do sistema

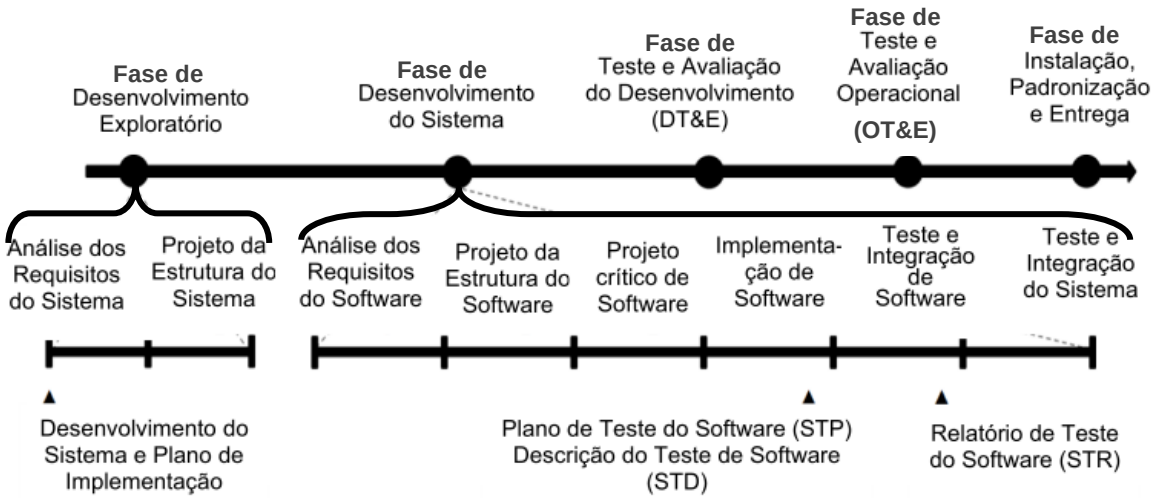
Geralmente projetistas e desenvolvedores de sistemas de combate não possuem uma forte mentalidade de segurança cibernética não são orientados a desenvolverem um sistema robusto a ciberataques. Tais projetistas, possuem uma forte mentalidade no desempenho que esses sistemas devem atender em situações de guerra [16].

Como as falhas cibernéticas ocorrem muitas vezes durante a implementação e desenvolvimento do sistema [36], as vulnerabilidades cibernéticas podem ser identificadas por meio de testes de segurança [37] e [38] de maneira a serem tratadas e solucionadas ainda durante esta etapa.

A Figura 3, apresenta as fases tradicionais do ciclo de vida de desenvolvimento de um sistema divididas em:

- Fase de Desenvolvimento Exploratório;
- Fase de Desenvolvimento do Sistema;
- Fase de Teste e Avaliação do Desenvolvimento;
- Fase de Teste e Avaliação Operacional; e
- Fase de Instalação, Padronização e Entrega.

Figura 3 - Fases e etapas do desenvolvimento do sistema de combate



Fonte: [14]

Observa-se na Figura 3 que as duas primeiras fases, Desenvolvimento Exploratório e Desenvolvimento do Sistema, podem ser subdivididas em etapas que estão representadas logo abaixo das fases na figura.

Na fase de Desenvolvimento Exploratório, a etapa de Análise dos Requisitos do Sistema é iniciada pelo Desenvolvimento do Sistema e Plano de Implementação. A etapa é seguida pelo Projeto da Estrutura do Sistema durante ainda a fase de Desenvolvimento Exploratório.

Na fase de Desenvolvimento do Sistema, tradicionalmente, são realizados os testes do software, após a etapa de Implementação de Software. O Teste e Integração de Software encerra os testes com o Plano de Teste do Software (STP, do inglês Software Test Plane, da última linha na figura) e gerado o Relatório de Teste do Software (STR, do inglês Software Teste Record, da última linha na figura).

É importante conhecer os requisitos e critérios do teste de segurança desde a fase inicial, onde o sistema está sendo idealizado, pois isso evitará falhas de segurança [16].

Nessa prática tradicional, em que a aplicação do teste de segurança é realizada no final do desenvolvimento do software [16], o que é uma limitação do ponto de vista de integração de sistemas.

Deve ser observada uma preocupação quanto aos executores dos testes, segundo a obra "*Surface Navy Combat Systems Engineering Strategy*" feita por Kathy Emery [14], eles devem ser conduzidos por uma instituição supervisora de P&D na fase de Desenvolvimento do Sistema durante as etapas de Implementação de Software, Teste e Integração de Software, e Teste e Integração de Sistema, conforme a Figura 3. Os testes principais para aceitação serão conduzidos nas fases seguintes a Fase de Desenvolvimento com as Fase de Teste e Avaliação do Desenvolvimento (DT&E, do inglês *Development Tests and Evaluation*) e na Fase de Teste e Avaliação Operacional (OT&E, do inglês *Operational Tests and Evaluation*).

Todos os testes deverão ser conduzidos pela instituição supervisora de P&D e por militares especializados na área técnica até a Fase de Teste e Avaliação do Desenvolvimento, enquanto na Fase de Teste e Avaliação Operacional os testes deverão ser executados por militares das áreas técnica e operacional [14].

4.2 Teste de segurança aplicado durante o desenvolvimento do software

Os mecanismos de teste de segurança garantem a qualidade do sistema e reduzem o risco e o custo de desenvolvimento [39]. Por meio de testes funcionais pode-se validar se os requisitos de segurança são adequadamente atendidos [38]. A Tabela 5 foi extraída da obra "*Security Testing: A Survey*" de Felderer [40], nela é citada algumas técnicas que podem ser usadas nos testes de segurança do software e que estão divididas conforme as características do ciclo de vida de desenvolvimento de um sistema de software seguro.

Tabela 5 - Técnicas do teste de segurança em um ciclo de vida de desenvolvimento de software

Técnicas do teste de segurança	**Descrição**	**Fases do ciclo de vida de desenvolvimento de software seguro**
Teste de segurança baseado em modelo	Baseado nos requisitos e no modelo de design criado durante a análise e o design	Análise & Projeto
Teste baseado em código e análise estática	Baseado na análise e no código-fonte e código de byte criado	Desenvolvimento
Teste de inserção e análise dinâmica	Executado em sistemas em execução ou em ambiente de produção da perspectiva do invasor	Implantação & Operação
Teste de regressão da segurança	Conduzido quando ocorrem mudanças no código de software existente	

Fonte: [40]

O primeiro teste da Tabela 5 é o teste de segurança baseado em modelo. Este teste verifica os requisitos de software relacionados às propriedades de segurança (confidencialidade, integridade, disponibilidade, autenticação, autorização e aceitação ou não repúdio) para definir o modelo de projeto de software através de critérios de filtro e evidências. Mais detalhes podem ser vistos nas referências [38] e [40].

O teste baseado em código e análise estática é usado para analisar e revisar o código-fonte do programa para identificar manualmente, ou automaticamente,

vulnerabilidades no software durante o desenvolvimento do código [16] e [40]. Algumas ferramentas automatizadas que verificam apontam as deficiências de proteção cibernética como ARCHER, Nmap, LAPSE, Nessus, MOPS, Coverity Scan, entre outros, podem ser encontradas nas referências [22], [41] e [42].

No teste de inserção e análise dinâmica, um operador, chamado de "testador", executa a análise em um sistema em execução e o compromete enviando uma carga útil da perspectiva de um invasor cibernético que possui informações insuficientes sobre o sistema de destino [43]. O invasor utiliza ferramentas manuais e automáticas para identificar vulnerabilidades [44]. O operador utiliza o teste "fuzzing[17]" que identifica possíveis vulnerabilidades de segurança, enviando dados válidos ou inválidos para o sistema [45].

Por fim, o teste de regressão de segurança garante que qualquer alteração no sistema, após seu desenvolvimento, não prejudique sua segurança [46]. Este teste é um processo importante no ciclo de vida de desenvolvimento do software, pois as alterações no código-fonte ocorrem com frequência devido a correções, patches, aprimoramentos ou ocorrência de dados corrompidos. Ao modificar o código existente, o "testador" deve realizar testes de segurança para validar se novos bugs de segurança foram introduzidos [47].

Tais testes de segurança devem ser incluídos nas etapas de Desenvolvimento de Software de Sistemas que foi ilustrado na Figura 3.

4.2.1 Implementação da segurança na fase de desenvolvimento

Esta seção especifica alguns critérios para teste de segurança do software dos Sistemas Navais, especialmente o sistema de combate conforme descrito no trabalho [16].

[17]Teste "*fuzzing*" - é uma técnica que consiste no envio de sequências de mensagens válidas e inválidas para um sistema, a fim de determinar possíveis causas de vulnerabilidades de segurança. Uma característica importante do "*fuzzing*" é que ele não requer conhecimento dos detalhes de implementação do sistema alvo. Esta técnica é muito útil para testar ataques de injeção, por exemplo, e pode ser combinada com outros mecanismos de teste [44].

Na Tabela 6, as categorias de implementação de segurança do software estão relacionadas com os requisitos de segurança do projeto do software: verificação e apresentação dos dados de entrada, funcionalidades de segurança, tempo e estado, processamento de erros, erro de código, encapsulamento e uso indevido da interface de programação de aplicativos, compreendem em 47 itens do "Guia para Segurança do Desenvolvimento de Software" [48].

Dessa forma, o trabalho do guia [48] permitiu compilar na Tabela 6 formas de mitigar as vulnerabilidades e definir os requisitos de segurança, desde o estágio inicial de desenvolvimento do software, o que permite sua utilização no desenvolvimento de softwares para sistemas navais como o sistema de combate [16].

Tabela 6 - Categorias de implementação da segurança do software na fase de desenvolvimento

CATEGORIA	SUBITENS	DESCRIÇÃO
1. Verificação e apresentação dos dados de entrada	• Injeção de *Structured Query Language* (SQL) • *Cross-site scripting* (XSS) • Divisão de resposta *HyperText Transfer Protocol* (HTTP) • Upload irrestrito de arquivos com forma perigosa • Manipulação de pacotes e injeção de recursos • Injeção de XQuery • Injeção XPath • Injeção de *Lightweight Directory Access Protocol* (LDAP) • Injeção de comando do sistema operacional • Redirecionamento URL para site não confiável • Falsificação de solicitação entre sites (CSRF) • Separação de respostas HTTP • *Integer overflow* • Inserção de string de formato não controlada • *Memory buffer overflow* • Inserção de string de formato não controlada • A dependência em entradas não confiáveis no domínio de segurança	Isso verifica a validade dos dados de entrada do usuário (ou programa) e os trata adequadamente em caso de falha.
2. Funcio-nalidades de Segurança	• Permissão para acessar funções significativas sem autenticação devida • Autenticação inadequada • Atribuição de permissão incorreta para recurso crítico • Uso de algoritmo de criptografia fraco • Uso de números aleatórios insuficientes	Isso é aplicado a política de autenticação, controle de acesso, gerenciamento de autoridade, senha e assim por diante.

	• Uso de comprimento de chave de criptografia inadequado • Armazenamento em texto simples de informações importantes • Uso de números aleatórios insuficientes • Senha codificada • Transmissão de texto simples de informações significativas • Chave de criptografia codificada • Uso de senhas vulneráveis • Divulgação de informações devido a cookies salvos no *Hard Disk Drive* (HDD) do usuário • Informações do sistema no comentário do código-fonte • Download de código sem verificação de integridade • Uso de *hash* unidirecional sem adicionar valor aleatório a senha antes de calcular o *hash* • Ausência de limitação de tentativas de autenticações repetitivas	
3. Tempo e estado (*Time and status*)	• Tempo de verificação, condição de corrida de tempo de uso • Loop com condição de saída inacessível ou função recursiva	Isso evita falhas de segurança causadas pelo gerenciamento inadequado de tempo e estado em sistemas ou ambientes paralelos em que mais de um processo está sendo executado.
4. Processamento de erros	• Divulgação de informações por meio de mensagem de erro • Detecção de uma condição de erro sem ação • Processamento inadequado para condições incomuns ou excepcionais	Isso evita que informações importantes vazem devido ao tratamento inadequado de erros.
5. Erro de código	• *Null poiter deference* • Use após o recurso liberado • Encerramento ou liberação imprópria de recursos • Uso de uma variável não inicializada	Isso evita falhas de segurança causadas por erros de codificação que os desenvolvedores podem cometer.
6. Encapsulamento	• Divulgação de dados devido a sessão errada • Código de depuração restante • Exposição de dados do sistema a controle não autorizado • Campo do tipo-array privado retornado de um método público • Dados públicos atribuídos ao campo do tipo-array privado	Isso evita a divulgação de informações e problemas de autoridade que surgem do encapsulamento insuficiente de dados ou funcionalidades confidenciais.
7. Uso indevido da interface de programação de aplicativos	• Decisão de segurança baseada em consulta no *Domain Name System* (DNS) • Uso de interface de programação de aplicativos vulneráveis	Isso evita o uso de interface de programação de aplicativo vulnerável.

Fonte: [48]

4.3 Uma proposta para o teste de segurança do software do sistema de combate

Os testes de segurança do software do sistema de combate de um navio de guerra devem ser realizados não apenas na camada de aplicação, mas também no *middleware* e no sistema operacional para aumentar a segurança cibernética do sistema como um todo [16].

As orientações básicas para melhorar os testes de segurança são:

1. Estabelecer categorias de testes de segurança do ponto de vista da segurança cibernética, de forma detalhada que possa satisfazer as propriedades de segurança [16]; e

2. Otimizar os subitens de teste de segurança, descritas na Tabela 6 da seção 4.1.1, de forma a eliminar as vulnerabilidades de segurança com base nas categorias de segurança mencionadas em [48].

Quando um software do sistema de combate é desenvolvido, o teste de segurança do aplicativo é tratado como menos importante do que o seu desempenho ou a funcionalidade [16], o que resulta nas seguintes limitações:

1. Não existe uma estratégia sistemática e específica definida para testes de segurança do aplicativo que podem ser utilizadas no sistema de combate após o estágio de desenvolvimento [16].

Por isso, a Tabela 6 se concentra principalmente em sistemas gerais de TI e o software do sistema de combate requer testes de segurança otimizados. Para reforçar fundamentalmente a segurança cibernética do software do sistema de combate, testes de segurança específicos e práticos devem ser conduzidos e podem ser aplicados durante o estágio de implementação do software [16].

4.4 Teste de segurança aprimorado para a fase de desenvolvimento por camadas

É ilustrado na Figura 4 um aprimoramento das categorias da Tabela 6 de [45] por [16], no ponto de vista da segurança cibernética, estabeleceu-se oito aprimoramentos categóricos: controle de dados de entrada, autenticação, controle de acesso, autorização, controlar erros de codificação, confidencialidade, controle de serviço e controle de log. O controle de acesso, controle de serviço e controle de log são considerados propriedades críticas de segurança [16].

Figura 4 - Categorias de teste de segurança aprimoradas para o software do sistema de combate

Fonte: [16]

A Tabela 7 apresenta a descrição de cada categoria de testes e as camadas de softwares correspondentes que compõem o sistema de combate. Consequentemente, os testes de segurança foram sistematicamente detalhados e

podem se tornar o padrão para prover a segurança cibernética necessária aos Sistemas Navais [16].

Tabela 7 - Categorias do teste de segurança do software

Categoria	Descrição	Camadas do Software		
		Aplicação	Middleware	Sistema Operacional
1. Controle de Entrada de Dados	• Deve ser verificado se dados de entrada inadequados no código-fonte são explorados; • Deve-se verificar se existem vulnerabilidades como inundação de dados.	✓		
2. Autenticação	• Deve ser verificado se a autenticação apropriada para funções significativas ou participantes do domínio é realizada; • Deve-se verificar se a conta e senha para login não são vulneráveis.	✓	✓	✓
3. Controle de Acesso	• Controles de acesso apropriados para recursos críticos ou participantes do domínio devem ser verificados; • O acesso à rede com base na porta IP ou o uso de comandos do sistema operacional com impacto importante no sistema deve ser verificado.	✓	✓	✓
4. Autorização	• Deve ser verificado se as permissões apropriadas são concedidas para recursos críticos (por exemplo, arquivos, diretórios).	✓		✓
5. Controlar Erros de Codificação	• Deve ser verificado se uma exposição de informação vulnerável ou status impróprio está acontecendo devido a um erro do desenvolvedor ou tratamento inadequado de erros.	✓		
6. Confidenciali-dade	• Deve-se verificar se informações significativas estão expostas devido à falta de criptografia, uso impróprio da chave de criptografia ou informações críticas embutidas no código-fonte.	✓	✓	

7. Controle de Serviço	• Deve ser verificado se serviços desnecessários (por exemplo, SNMP, FTP, finger, r-command) e funções (por exemplo, pasta compartilhada e processador de texto no Windows) que podem ser explorados estão ativados no sistema operacional; • Deve-se verificar se as funções de segurança obrigatórias (por exemplo, firewall do console no Windows) estão configuradas.		✓
8. Controle de Log	• Deve-se verificar se informações importantes de registro são geradas e gerenciadas para análise pós-acidente cibernético.	✓	✓

Fonte: [16]

De modo a refinar e detalhar o teste de segurança e atender as necessidades para a realização do teste nos Sistemas Navais Embarcados, foi estabelecido na Tabela 7, pelos autores de [16] tabelas específicas e aprimoradas para a camada de aplicação, do *middleware* e do sistema operacional, que serão apresentadas nas tabelas 8, 9 e 10 respectivamente.

Tabela 8 - Subitens do teste de segurança na camada de aplicação

Categoria	Subitens
1. Controle Entrada de Dados (6 itens)	• Manipulação de pacotes e injeção de recursos; • Inserção de string de formato não controlada; • Confiança em entrada não confiável uma decisão de segurança; • Injeção de comando do sistema operacional; • *Integer overflow*; e • *Memory buffer overflow*.
2. Autenticação (4 itens)	• Permissão para acessar funções significativas sem autenticação devida; • Autenticação inadequada; • Uso de senha vulnerável; e • Uso de *hash* unidirecional sem adicionar valor aleatório a senha antes de calcular o *hash*
3. Controle de Acesso (1 item)	• Controle de acesso a função tática significativa (arma, gerenciamento e controle de pouso e decolagem, etc.)
4. Autorização (1 item)	• Atribuição de permissão incorreta para recurso crítico (arquivo executável, arquivo de configuração, biblioteca, diretório, etc.)
5. Controlar Erros de	• Download de código sem verificação de integridade; • Tempo de verificação, condição de corrida de tempo de uso; • Loop com condição de saída inacessível ou função recursiva;

Codificação (16 itens)	• Divulgação de informações por meio de mensagem de erro; • Detecção de condição de erro sem ação; • Processamento inadequado para condições incomuns ou excepcionais; • *Null pointer deference;* • Encerramento ou liberação imprópria de recursos; • *Use after freed resource;* • Uso de uma variável não inicializada e resto de código de depuração; • Divulgação de dados devido a sessão errada; • Exposição de dados do sistema a um controle não autorizado; • Campo do tipo-array privado retornado de um método público; • Dados públicos atribuídos ao campo do tipo-array privado; e • Uso de API vulnerável.
6. Confidenci-alidade (8 itens)	• Senha codificada no código-fonte; • Chave de criptografia embutida no código-fonte; • Divulgação de informações significativas do sistema (ID, senha, etc.) por meio de comentários; • Uso de algoritmo de criptografia fraco; • Uso de comprimento de chave de criptografia inadequado; • Armazenamento de texto simples de informações significativas; • Transmissão de texto simples de informações significativas; e • Uso de números aleatórios insuficientes.
8. Controle de Log (1 item)	• Geração e armazenamento de log do autor relacionado a dados significativos de segurança (log-in, acesso, processamento de informações táticas, etc.).

Fonte: [16]

Tabela 9 - Subitens do teste de segurança na camada do *middleware*

Categoria	Subitens
2. Autenticação	Autenticação apropriada para participantes do domínio
3. Controle de Acesso	Controle de acesso para os participantes do domínio
6. Confidencialidade	Criptografia para a mensagem transmitida

Fonte: [16]

Tabela 10 - Subitens do teste de segurança na camada do sistema operacional

Categoria	Subitens
2. Autenticação (6 itens)	• Excluir conta padrão e desnecessária; • Verificação de senha vulnerável (use uma combinação de números, caracteres, caractere especial); • Verificação de segurança de login [Windows]; • Controle da conta de administrador e convidado [Windows]; • Controle de contas anônimas no grupo de administradores [Windows]; e • Controle de acesso remoto à conta "root".
3. Controle de Acesso (4 item)	• Controle do uso do comando "su" por usuários comuns; • Controle de IP e porta acessíveis; • Gerenciamento de segurança de acesso remoto [Windows]; e • Controle de acesso remoto ao arquivo SAM (gerenciamento da conta de segurança) [Windows].
4. Autorização (8 item)	• Configurando UMASK, PATH para arquivos; • Configurando o privilégio de acesso à página inicial do usuário e aos

	diretórios significativos; • Verifique se os arquivos normais foram deletados no diretório dev/; • Verifique se arquivos e diretórios sem proprietário foram excluídos; • Definição de proprietário e privilégio para arquivos significativos do sistema; • Verifique se existem arquivos com configurações SUID e GUID desnecessárias; • Não permitir enumeração anônima de contas SAM e compartilhamentos [Windows]; e • Gerenciamento de segurança de diretório
7. Controle de Serviço (11 itens)	• Verifique os serviços desnecessários (SNMP, ftp, etc.); • Definir proprietário e privilégio para arquivos "cron"; • Parada de serviços NFS/RPC ou configuração de segurança; • Exclua a pasta compartilhada e configure o firewall do console [Windows]; • Definindo stop para "*daemon*" vulnerável (finger, r-commands, exec, etc.) no arquivo inetd.d; • Interrompa trabalhos de agendamento desnecessários; • Exclua aplicativos desnecessários (MS-Office, processador de texto, etc.) [Windows]; • Configurando o registro para defender o ataque DoS (*Denial-of-service attack*, expressão do inglês) [Windows]; • Restringir função de suporte remoto [Windows]; e • Verifique se os programas de segurança do console (controle USB, NAC, etc.) estão instalados [Windows].
8. Controle de Log (1 item)	• Definição de armazenamento significativo de log do sistema

Fonte: [16]

Observa-se que as três tabelas possuem itens em comum com a Tabela 6, porém estão concentradas nos itens críticos de segurança, controle de acesso, controle de serviço e controle de log (ou registro). Esses testes não são exaustivos e podem ser aperfeiçoados com o desenvolvimento de novas técnicas de segurança cibernética.

Vale ressaltar que tais técnicas utilizadas no teste de segurança proposto não contempla um teste em *firmwares*, pois para essa camada do hardware são necessárias ferramentas específicas que permitirão a investigação de vulnerabilidades que não estão enquadradas nos requisitos do teste de segurança proposto.

Esse fato é devido a grande maioria esmagadora das vulnerabilidades em *firmwares* estarem presente nas diversas versões do kernel encontrados nos dispositivos de hardware dos fabricantes [24].

4.5 Propostas de testes de segurança durante os processos de desenvolvimento através do modelo em V

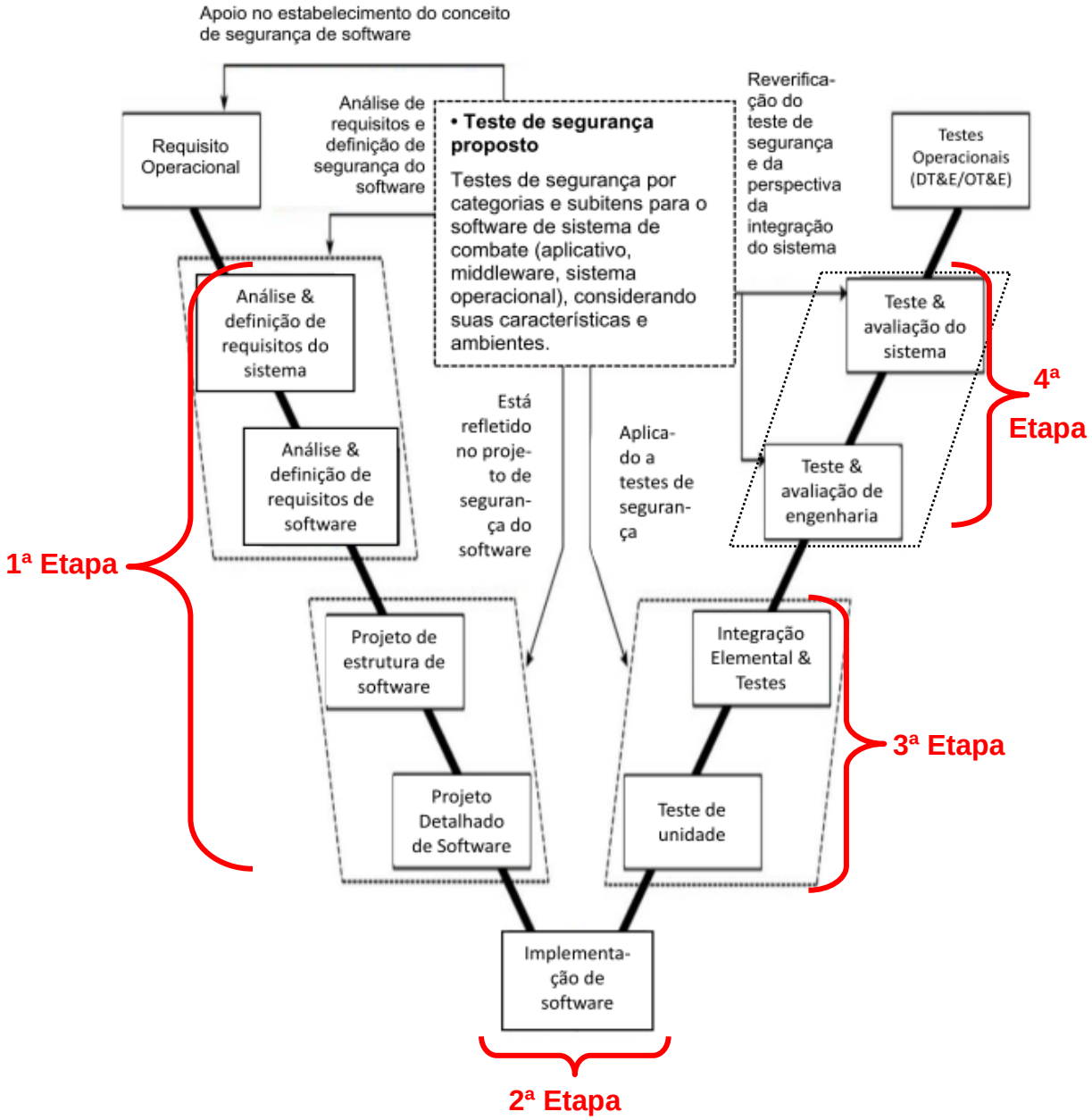
O modelo em V fornece uma ilustração útil das atividades de engenharia de sistemas, incluindo o desenvolvimento de softwares, durante os estágios do ciclo de vida. No modelo em V, o tempo e a maturidade do sistema procedem da esquerda para a direita. O núcleo do “V” descreve a linha de base em evolução, desde o acordo de requisitos do usuário até a identificação de um conceito de sistema e a definição dos componentes do sistema que constituirão o produto final. Com o tempo se movendo para a direita e com a maturidade do sistema mostrada verticalmente, a linha de base em evolução define o lado esquerdo do núcleo do “V” conforme a Figura 5 [49]. Isso possibilita uma rastreabilidade do modelo, facilitando a resolução de problemas durante o projeto e se necessário a revisão ou mudança de alguns requisitos.

Tal modelo mostra a visão dos processos de análise, desenvolvimento e integração do software do sistema, representados pelos quadrados dos lados esquerdo e direito do modelo em V.

Nesse modelo, ilustrado na Figura 5, fica evidente que devem ser estabelecidos os conceitos de segurança de software durante os Requisitos Operacionais, antes da própria análise dos requisitos e da definição dos critérios de segurança.

O teste de segurança pode ser dividido em quatro etapas, conforme indicado no modelo da Figura 5 e aprimorado pelos autores da obra [16], relacionando-as com as técnicas do teste de segurança da Tabela 5, conforme detalhados abaixo dessa figura:

Figura 5 - Processo do teste de segurança proposto no modelo em V do desenvolvimento do sistema de combate.



Fonte: [14]

1ª Etapa: ocorre ao longo da Análise e definição de requisitos do sistema até o Projeto Detalhado de Software. Nessa etapa, deverá ser realizado o teste de segurança baseado em modelo, conforme descrito na Tabela 5, a fim de verificar se os requisitos de software estão relacionados as propriedades de segurança (confidencialidade, integridade, disponibilidade, autenticação, autorização e não aceitação ou não repúdio) indicados nas referências [38] e [40];

2ª Etapa: é executada na fase de desenvolvimento, durante a Implementação do Software (representado pela base do modelo em V), a qual deverão ser realizados testes baseados em código e análise estática, conforme retratado na Tabela 5. Tais testes foram exemplificados na seção 4.2, deste trabalho, e nas referências [22], [41] e [42];

3ª Etapa: realiza-se os testes de inserção e de análise dinâmica, conforme os subitens das categorias de segurança aprimoradas por [16] constantes nas Tabelas 8, 9, 10 e nas referências [43], [44], [45], [46] e [47]. Nessa etapa já existe uma integração elemental entre o aplicativo do sistema de combate, *middleware* e o sistema operacional, por isso o uso das tabelas aqui nessa etapa citada; e

4ª Etapa: o teste de regressão de segurança será aplicado durante o Teste e avaliação de engenharia e Teste e avaliação do sistema, conforme indicado na Figura 5, a fim de verificar se as alterações do código-fonte advindas da integração dos sistemas não geraram uma nova vulnerabilidade [46] e [47].

A abordagem proposta para testes de segurança de software concentra-se na implementação de software e, principalmente, no teste e integração do sistema, conforme apresentado na Figura 5.

Diferente do desenvolvimento tradicional, da Figura 3, há funções e vantagens dos testes de segurança no processo de desenvolvimento que podem ser mais bem compreendidas no que é oferecido pelo modelo em V, como por exemplo, a reverificação do teste de segurança na perspectiva da integração do sistema [16], o que mostra que nesse ponto de vista esse teste não é limitado.

Esse método facilita o desenvolvimento sistemático de sistemas, que pode ser utilizado em sistemas com as características de software do sistema de combate, combinando os requisitos, a análise do sistema e o processo de design no lado esquerdo com o processo de teste e avaliação no lado direito. Porém, no caso do sistema de combate que é um sistema de missão crítica e processa dados táticos envolvendo inimigos, tudo ocorrendo em tempo real, o teste de segurança não pode ter precedência sobre os testes de desempenho e função [16].

4.6 Melhoria dos testes com uso de ferramentas para sistemas de código aberto Linux

Baseado na obra [16] e no “Guia de Segurança - Conceitos e técnicas para proteger servidores e estações de trabalho *Red Hat Enterprise Linux* (RHEL, sigla do inglês)” [22], este trabalho irá expor métodos e ferramentas que podem ser utilizados em sistemas Linux.

A plataforma Linux foi escolhida nesse trabalho pois é um sistema operacional de código-fonte aberto, o que beneficia a MB por ser manipulável e ter baixo custo no ponto de vista de aquisição do sistema operacional.

Abaixo serão exibidas ferramentas que estão de acordo com as Tabelas 8, 9 e 10, desta obra, e auxiliarão na detecção de vulnerabilidades nas camadas do sistema operacional, *middleware* e do aplicativo, bem como na rede local onde os dados dos sistemas estão disponíveis:

- O “NMAP” é um aplicativo que mapeia todos os hosts da rede, ele é uma boa base para estabelecer uma política de utilização de serviços seguros e restringir serviços não utilizados [22];
- A ferramenta “Nessus” realiza análise dinâmica de código que oferece relatórios completos, verificação de host e pesquisas de vulnerabilidades em tempo real, mas pode haver falsos positivos e falsos negativos, nesse caso o operador deve verificar cada caso em particular [22];
- O “OpenVAS” é outro instrumento de análise dinâmica que possui outras ferramentas embutidas baseada em web, desktop e linha de comando para controlar vários componentes de solução [22];
- O “Nikto” é um excelente scanner de script de *common gateway interface* (CGI). Não verifica apenas vulnerabilidades de CGI, mas também o faz de maneira evasiva, para escapar dos sistemas de detecção de intrusões [22];
- O “Coverity Scan” é um recurso de análise estática de código e basicamente busca e corrige defeitos em codificações de projetos de fonte aberta escritos em Java, C/C++, C#, JavaScript, Ruby ou Phyton [50]; e
- O “Veracode” é um conceito de aplicação que atua em cinco pontos críticos, no que tange o desenvolvimento seguro do software: segurança no desenvolvimento

do ciclo de vida do software, educação em segurança, cadeia de suprimentos segura, monitoramento de ataques em redes [51].

No guia [22], em seu capítulo 4, é possível encontrar ferramentas e serviços adicionais como:

- LibreSWAN é uma ferramenta poderosa para a criação de VPNs seguras em ambientes Linux e é amplamente usada em muitos cenários de rede onde a segurança é uma prioridade [22];
- OpenSSL fornece uma implementação de código aberto das camadas de protocolo SSL/TLS (*Secure Sockets Layer / Transport Layer Security*, do inglês) para comunicações seguras pela Internet, além de ser utilizado para criptografar dados transmitidos pela rede, como em conexões HTTPS, SMTPS (para e-mails seguros), FTPS (para transferências de arquivos seguras) [22];
- STUNNEL adiciona criptografia SSL/TLS a serviços de rede que originalmente não suportam criptografia. Isso inclui aplicativos como servidores POP3 (servidor unidirecional no qual o email é recebido e mantido no servidor de email), IMAP (servidor de e-mails recebidos), SMTP (servidor de e-mails enviados), servidores VNC (VNC, do inglês *Virtual Network Computing*), bancos de dados MySQL, entre outros. Protegendo as comunicações de rede entre clientes e servidores, garantindo que os dados sejam transmitidos de forma segura [22];
- *Advanced Intrusion Detection Environment* (AIDE, sigla do inglês) é uma ferramenta de código aberto de detecção de intrusão que é usada para monitorar a integridade dos arquivos em sistemas de distribuição Linux. Detecta alterações não autorizadas em arquivos de sistema e em suas configurações [22]; e
- USBGUARD protege, monitora e facilita a auditoria do sistema contra ameaças de segurança relacionadas ao uso de dispositivos USB, como pen drives, teclados USB, discos rígidos externos e outros dispositivos de armazenamento USB que podem representar riscos à segurança [22].

O conhecimento da utilidade de tais ferramentas possibilitou criar a Tabela 11 que associa as ferramentas às Etapas citadas na seção 4.5 a fim de identificar em quais momentos elas podem ser empregadas por profissionais de segurança (técnico de segurança de sistemas) ou desenvolvedores de sistemas.

A referência [37] cita algumas ferramentas *open-source* adicionais que serão incluídas na tabela abaixo.

Tabela 11 – Associação de softwares de identificação de vulnerabilidades com as Etapas do teste de segurança ao longo do ciclo de vida de desenvolvimento do sistema.

Tipo de análise	Ferramentas disponível	Etapa do ciclo de vida para utilização da ferramenta	Profissional envolvido
Análise estática de Código-Fonte	*Coverity Scan, C Code Analyser, LAPSE, ASTREE e RATS*	2ª Etapa	Desenvolvedor
Monitoramento de Vulnerabilidades de redes	*NMAP, Nessus, Veracode e SuperScan*	3ª e 4ª Etapas / Operação do Sistema	Desenvolvedor / Profissional de segurança
Monitoramento de vulnerabilidades de aplicações Web	*Nikto, OpenVAS, OWASP, Spike Proxy e EOR*	3ª e 4ª Etapas / Operação do Sistema	Desenvolvedor / Profissional de segurança
Análise dinâmica de Código-Fonte	*Nessus, NProf, Foundstone .NETMon e CLR Profiler*	3ª e 4ª Etapas / Operação do Sistema	Desenvolvedor / Profissional de Segurança
Ferramenta de análise de Configurações	*Foundstone SSLDigger, PermCalc e USBGUARD*	4ª Etapa / Operação do Sistema	Desenvolvedor / Profissional de Segurança

Fonte: próprio autor

Essas ferramentas *open-source* podem ser utilizadas pela Marinha do Brasil para identificar vulnerabilidades em seus sistemas já desenvolvidos ou em desenvolvimento.

Sugere-se que a Diretoria de Comunicações e Tecnologia da Informação da Marinha (DCTIM) realize a análise de tais ferramentas e faça uma homologação daquelas que serão de utilidade para os setores da Marinha que tenham envolvimento com a segurança cibernética.

5 CONCLUSÃO

Este trabalho acadêmico conseguiu relacionar os ataques cibernéticos ao ambiente marítimo e mostrou que os sistemas estão desprotegidos e apresentam

vulnerabilidades. Além dos sistemas encontrados em navios civis, os sistemas de combate de um navio de guerra não estão devidamente protegidos e apresentam vulnerabilidades que foram tratadas no desenvolvimento desta pesquisa.

Diante dessas ameaças, este trabalho evidenciou a importância da proteção cibernética de Sistemas Navais de Embarcações Militares, através da proposta de um teste de segurança para tais sistemas de forma a evitar ou amenizar ataques cibernéticos, e orientar os desenvolvedores dos sistemas quanto a uma estrutura de segurança robusta.

Por fim, este trabalho apresentou uma metodologia para a realização de um teste de segurança desde a sua concepção até a sua entrega, permitindo incrementar a proteção cibernética dos Sistemas Navais de Embarcações Militares.

Para a realidade da Marinha do Brasil, a supervisão dos testes deverá ocorrer por uma instituição de P&D, logo sugere-se o Instituto de Pesquisas da Marinha (IPqM) [52]. Para a tarefa de testes e avaliação operacional sugere-se o Centro de Análises de Sistemas Navais (CASNAV), por tem como missão contribuir para o desenvolvimento científico e tecnológico da MB e do país [53].

Sugere-se a criação de uma norma técnica para Sistemas Digitais Operativos, diferente dos sistemas digitais que são homologados pela DCTIM, que geralmente são de cunho administrativo. Dessa forma uma organização especializada da MB, como o Centro de Apoio a Sistemas Operativos (CASOP), possa definir os testes de segurança para os sistemas navais de embarcações militares de forma sistemática e específica.

Visando a proteção do software em SDO em desenvolvimento nos projetos de sistemas navais, sugere-se que o CASOP e o CASNAV realizem as práticas demostradas pelo modelo em V proposto utilizando as técnicas e ferramentas citadas neste trabalho.

Sugere-se verificar na prática, em trabalhos futuros, a efetividade das técnicas, ferramentas e métodos mencionados neste trabalho, comparando resultados em projetos práticos da MB, como por exemplo o sistema de combate das Fragatas Classe “Tamandaré” ou outro Sistema Naval como por exemplo o SisGAAz.

REFERÊNCIAS

1 Conheça 5 vantagens da cadeia de suprimentos digital. **Orbit Logistics**, 08 jun. 2022. Disponível em: https://blog.portalvmi.com.br/conheca-5-vantagens-da-cadeia-de-suprimentos-digital/. Acesso em 23 set. 2023.

2 KOCH, Robert; GOLLING, Mario. Weapons systems and cyber security-a challenging union. In: **2016 8th International Conference on Cyber Conflict (CyCon)**. IEEE, 2016. p. 191-203. DOI: https://doi.org/10.1109/CYCON.2016.7529435. Acesso em: 26 jul. 2023.

3 OLADIMEJI, Saheed; KERNER, Sean Michael. **SolarWinds hack explained:** Everything you need to know. 27 jun. 2023. Disponível em: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know. Acesso em: 11 ago. 2023.

4 PETRONI, Benedito Cristiano; GLORIA JUNIOR, Irapuan; GONÇALVES, Rodrigo Franco. SISTEMAS CIBER-FÍSICOS. **SACOMANO, JB**, p. 47-55.

5 MARINHA DO BRASIL. Estado-Maior da Armada. **EMA-419:** Doutrina Cibernética da Marinha. Brasília, DF, 1. ed., 2021.

6 PROGOULAKIS, Iosif; ROHMEYER, Paul; NIKITAKOS, Nikitas. Cyber physical systems security for maritime assets. **Journal of Marine Science and Engineering**, v. 9, n. 12, p. 1384, 2021. DOI: https://doi.org/10.3390/jmse9121384. Acesso em: 18 jul. 2023

7 Allianz Risk Barometer - Results appendix 2023. **ALLIANZ**, jan. 2023, p. 17. Disponível em: https://commercial.allianz.com/content/dam/onemarketing/commercial/commercial/reports/Allianz-Risk-Barometer-2023-Appendix.pdf. Acesso em: 28 ago. 2023.

8 HASSAN, Nihad A. **Ransomware Revealed:** A Beginner's Guide to Protecting and Recovering from Ransomware Attacks. Berkeley: Apress, 2019. 218 p. ISBN 9781484242544. DOI: https://doi.org/10.1007/978-1-4842-4255-1. Acesso em: 12 jun. 2023.

9 Phishing e Outros Golpes. CERT. Disponível em: https://cartilha.cert.br/fasciculos/#codigos-maliciosos. Acesso em: 11 ago 2023.

10 Cadeia de suprimentos: o que é o também chamado supply chain?. Disponível em: https://www.umov.me/cadeia-de-suprimentos-e-supply-chain/#post-title1 Acesso em 13 nov. 2023.

11 MISTRAL an AXISCADES company. **Naval Systems**. Disponível em: https://www.mistralsolutions.com/defense-solutions-page/solutions/naval-systems/. Acesso em: 03 nov. 2023.

12 CARTER, Robert C. *et al*. **A Job Analysis of the Aegis Combat System Submodes**. NAVAL BIODYNAMICS LABORATORY. 1982.

13 BEN FARAH, Mohamed Amine et al. Cyber Security in the Maritime Industry: A Systematic Survey of Recent Advances and Future Trends. **Information**, v. 13, n.1, p. 22, 6 jan. 2022. Disponível em: https://doi.org/10.3390/info13010022. Acesso em 02 set. 2023.

14 EMERY, Kathy. Surface Navy Combat Systems Engineering Strategy. Program Executive Office - Integrated Warfare Systems, **Anais** [...], 4 mar. 2010. Disponível em: https://nps.edu/documents/103424733/107333295/IWS+OA+Briefing+to+NPS+March0410.pdf/db126956-3457-46cb-bd81-7fa14959105a. Acesso em: 28 ago. 2023.

15 United States Government Accountability Office. **GAO-19-128**: WEAPON SYSTEMS CYBERSECURITY: DOD Just Beginning to Grapple with Scale of Vulnerabilities. 09 out. 2018. Disponível em: https://www.gao.gov/assets/gao-19-128.pdf. Acesso em: 28 ago. 2023.

16 YI, Cheol-Gyu; KIM, Young-Gab. Security Testing for Naval Ship Combat System Software**. IEEE Access**, v.9, p.66839-66851, 30 abr. 2021. Disponível em: https://doi.org/10.1109/access.2021.3076918. Acesso em: 10 jul. 2023.

17 MARINHA DO BRASIL. Estado-Maior da Armada. **PEM 2040**: Plano Estratégico da Marinha. Brasília, DF, 2020. ISBN 978-65-991468-0-0. Disponível em: https://www.marinha.mil.br/pem2040. Acesso em: 02 set. 2023.

18 BRASIL. **Decreto nº 10.569, de 9 de dezembro de 2020**. Aprova a Estratégia Nacional de Segurança de Infraestruturas Críticas. Brasília, DF, 9 dez. 2020. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2020/decreto/D10569.htm. Acesso em: 02 set. 2023.

19 BRASIL, Ministério da Defesa. **MD-40-M-1:** MANUAL DE BOAS PRÁTICAS PARA A GESTÃO DO CICLO DE VIDA DE SISTEMAS DE DEFESA. 1. ed., 2019.

20 VOO, Julia; HEMANI, Irfan; CASSIDY, Daniel. National Cyber Power Index 2022. **CYBER PROJECT.** Harvard Kennedy School - Belfer Center for Science and International Affairs, Cambridge, MA, set. 2022. Disponível em: https://www.belfercenter.org/sites/default/files/files/publication/CyberProject_National%20Cyber%20Power%20Index%202022_v3_220922.pdf. Acesso em: 02 set. 2023.

21 ASCH, Richard Harold Geraldo. Integração de Sistemas de Combate – ISC. In: Apresentação do Power Point sobre a Plataforma de Combate – PARTE A, color. Curso de Aperfeiçoamento Avançado do Centro de Instrução Almirante Alexandrino. Rio de Janeiro 2023.

22 JAHODA, M. *et al*. **Red Hat Enterprise Linux 7 - Security Guide: Concepts and techniques to secure RHEL servers and workstations**. [s.l.] Red Hat, Inc., 2020. Disponível em: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/pdf/security_guide/red_hat_enterprise_linux-7-security_guide-en-us.pdf. Acesso em: 12 set. 2023.

23 O que é software open source?. **RedHat**, 11 mai. 2022. Disponível em: https://www.redhat.com/pt-br/topics/open-source/what-is-open-source-software. Acesso em: 18 ago. 2023.

24 FREITAS, Osmany Barros de; CORRÊA, Françoa Taffarel Rosário; SANTOS, Aldri Luiz dos; PEREIRA JUNIOR, Lourenço Alves. Caracterização das vulnerabilidades dos roteadores Wi-Fi no mercado brasileiro. *In*: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC), 41., 2023, Brasília/DF. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2023. p. 183-196. ISSN 2177-9384. DOI: https://doi.org/10.5753/sbrc.2023.487.

25 TERZIĆ, Aida; AKELJIĆ, Bekir. BASIC INPUT/OUTPUT SYSTEM BIOS FUNCTIONS AND MODIFICATIONS. **INTERNATIONAL UNIVERSITY TRAVNIK**, 08 nov. 2017.

26 ALVES, Paulo. **O que é boot no PC? Entenda o processo de inicialização**. TechTudo, 03 jun. 2021. Disponível em: https://www.techtudo.com.br/noticias/2021/06/o-que-e-boot-no-pc-entenda-o-processo-de-inicializacao.ghtml. Acesso em: 18 ago. 2023.

27 MACHADO, Raphael. **Segurança da Informação: Apostila da Disciplina**. Ed. Universidade Federal Fluminense - UFF, 2020.

28 Security Vulnerability Assessment Methodology for the Petroleum and Petrochemical Industries. **American Petroleum Institute – API**, ed. 2, Washington, DC, out. 2004. Disponível em: https://www.nrc.gov/docs/ML0502/ML050260624.pdf. Acesso em: 12 set. 2023.

29 e-Navigation traz mais segurança e aprimora a navegação. **Agência Marinha de Notícias**, 24 fev. 2022. Disponível em: https://www.marinha.mil.br/agenciadenoticias/e-navigation-traz-mais-seguranca-e-aprimora-navegacao. Acesso em: 06 set. 2023.

30 Guia de Aperfeiçoamento da Segurança Cibernética para Infraestrutura Crítica. **National Institute of Standards and Technology – NIST**. Tradução: Departamento de Comércio dos Estados Unidos, ed. 1.1, 2018. Título original: Framework for Improving Critical Infrastructure Cybersecurity. DOI: https://doi.org/10.6028/NIST.CSWP.04162018pt. Acesso em: 06 set. 2023.

31 BARKER, William C. et al. Gestão de risco de ransomware: Um perfil do Framework de segurança cibernética. **National Institute of Standards and Technology**. Tradução: Departamento de Comércio dos Estados Unidos, fev. 2022. DOI: https://doi.org/10.6028/NIST.IR.8374.por. Acesso em: 06 set. 2023.

32 Cyber security at sea: a brand new naval cyber framework. **THALES Group**. 22 set. 2022. Disponível em: https://www.thalesgroup.com/en/worldwide-defence-naval-forces/above-water-warfare/magazine/cyber-security-sea-brand-new-naval. Acesso em: 21 out. 2023.

33 STATE-OF-THE-ART. In: CAMBRIDGE dictionary. Disponível em: https://dictionary.cambridge.org/us/dictionary/english/state-of-the-art. Acesso em 21 out.2023.

34 United States Navy. **NSWCDD/MP-13/22:** Leading Edge: Combat Systems Engineering & Integration. Naval Surface Warfare Center, Dahlgren, Virginia, fev. 2013. Disponível em: https://www.navsea.navy.mil/Portals/103/Documents/NSWC_Dahlgren/LeadingEdge/CSEI/CombSys.pdf. Acesso em: 28 ago. 2023.

35 BAKKEN, David E. **Middleware:** What it is, and How it Enables Adapdvity and Dependability. 43º Meeting of IFIP Working Group 10.4. Sal, Cabo Verde, 4 jan. 2003. Disponível em: https://webhost.laas.fr/TSF/IFIPWG/Workshops&Meetings/43/01-Bakken.pdf. Acesso em: 11 ago 2023.

36 United States Department of Defense. KOSSIAKOFF, A. *et al*. **AD-A022 160:** DOD Weapon Systems Software Management Study. Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 01 jun. 1975. Disponível em: https://apps.dtic.mil/sti/citations/ADA022160. Acesso em: 28 ago. 2023.

37 TIAN-YANG, Gu; YIN-SHENG, Shi; YOU-YUAN, Fang. Research on Software security testing. **World Academy of Science, Engineering and Technology**, v. 70, n. 2009, p. 647–651, 2010. Disponível em: https://publications.waset.org/13580/pdf. Accesso em: 16 jul. 2023.

38 FELDERER, Michael *et al*. Model-based security testing: a taxonomy and systematic classification. **Journal of Software Testing, Verification And Reliability**, 24 jul. 2015. DOI: https://doi.org/10.1002/stvr.1580. Acesso em: 16 jul. 2023.

39 European Telecommunications Standards Institute**. ETSI EG 203 251:** Methods for Testing & Specification; Risk-based Security Assessment and Testing Methodologies. Sophia Antipolis, França, 2016.

40 FELDERER, Michael et al. Security Testing: A Survey. **Advances in Computers**, v. 101, Elsevier , Cambridge, MA, USA ,p. 1-51, 2016. DOI: https://doi.org/10.1016/bs.adcom.2015.11.003. Acesso em:16 jul. 2023.

41 CHESS, Brian; WEST, Jacob. **Secure Programming with Static Analysis**. Boston, MA. Pearson Education, 2007. ISBN 978-0-321-42477-8.

42 BRUCKER, Achim D.; SODAN, Uwe. Deploying static application security testing on a large scale. Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI). Anais...Wien, Österreich: Gesellschaft für Informatik, p. 91-101, 2014. DOI: https://dl.gi.de/handle/20.500.12116/20071. Acesso em: 16 jul. 2023.

43 SCARFONE, Karen *et al*. Technical guide to information security testing and assessment. **NIST Special Publication**, 2008. Disponível em: https://www.nist.gov/privacy-framework/nist-sp-800-115. Acesso em: 22 jun. 2023.

44 MATHEU-GARCÍA, Sara N. *et al*. Risk-based automated assessment and testing for the cybersecurity certification and labelling of IoT devices. **Computer Standards & Interfaces**, v. 62, p. 64-83, 2019. DOI: https://doi.org/10.1016/j.csi.2018.08.003. Acesso em: 16 jul. 2023.

45 GODEFROID, Patrice *et al*. SAGE: whitebox fuzzing for security testing. **Communications of the ACM**, v. 55, n. 3, p. 40-44, 2012. DOI: https://doi.org/10.1145/2093548.2093564. Acesso em: 16 jul. 2023.

46 FELDERER, Michael; FOURNERET, Elizabeta. A systematic classification of security regression testing approaches. **International Journal on Software Tools for Technology Transfer**, v. 17, n. 3, p. 305-319, 21 jan. 2015. DOI: https://doi.org/10.1007/s10009-015-0365-2. Acesso em: 16 jul. 2023.

47 WAHEED, Usman. **Security regression testing framework for Web application development**. 12 dez. 2014. Dissertação de Mestrado. Department of Informatics, University of Oslo, 2014.

48 Korea Internet & Security Agency – KISA e Ministry of the Interior and Safety. **Guia para segurança do desenvolvimento de software**. Tradução: Machine Translate, Google, 2021. Título original: 소프트웨어_개발보안_가이드. Disponível em: https://www.mois.go.kr/frt/bbs/type001/commonSelectBoardArticle.do;jsessionid=TjAX2IwVk6hpONx8dKSZ4VTj.node10?bbsId=BBSMSTR_000000000015&nttId=88956. Acesso em: 12 set. 2023.

49 HASKINS, Cecilia (ed.). **Systems engineering handbook: a guide for sytem life cycle processes and activities**. v. 3, Seattle, WA, jun. 2006.

50 Coverity Scan - Static Analysis. **SYNOPSYS**, Sunnyvale, CA, 2023. Disponível em: https://scan.coverity.com/. Acesso em: 06 set. 2023.

51 Intelligent Software Security. **VERACODE**, 2023. Disponível em: https://www.veracode.com/. Acesso em: 06 set. 2023.

52 Missão. Instituto de Pesquisas da Marinha do Brasil. Marinha do Brasil. Disponível em: https://www.marinha.mil.br/ipqm/missao. Acesso em: 21 out. 2023.

53 Missão e Visão. Centro de Análises de Sistemas Navais. Marinha do Brasil. Disponível em: https://www.marinha.mil.br/casnav/?q=node/65. Acesso em: 21 out. 2023.

ANEXO A

Neste anexo está exposto os critérios utilizados na obra [20] para chegar aos índices do poder cibernético. O poder cibernético é a implantação efetiva de capacidades cibernéticas por um estado para atingir seus objetivos nacionais.

Critério 1 - Vigilância e monitoramento de grupos domésticos: recursos de vigilância cibernética para monitorar, detectar e coletar informações sobre ameaças domésticas;

Critério 2 - Fortalecimento e aprimoramento das defesas cibernéticas nacionais: o país priorizou a defesa do governo, ativos de sistemas nacionais e a melhoria da higiene e resiliência cibernética nacional;

Critério 3 - Controlando e manipulando o ambiente de informação: O país espalha propaganda doméstica, cria e amplifica desinformação no exterior e usa recursos cibernéticos para atingir e interromper grupos estrangeiros;

Critério 4 - Coleta de inteligência estrangeira para segurança nacional: O país extraiu segredos nacionais de um adversário estrangeiro por meios cibernéticos;

Critério 5 - Crescente competência nacional em tecnologia cibernética e comercial: O país tentou aumentar sua indústria de tecnologia doméstica ou usou meios cibernéticos para desenvolver outras indústrias no país.

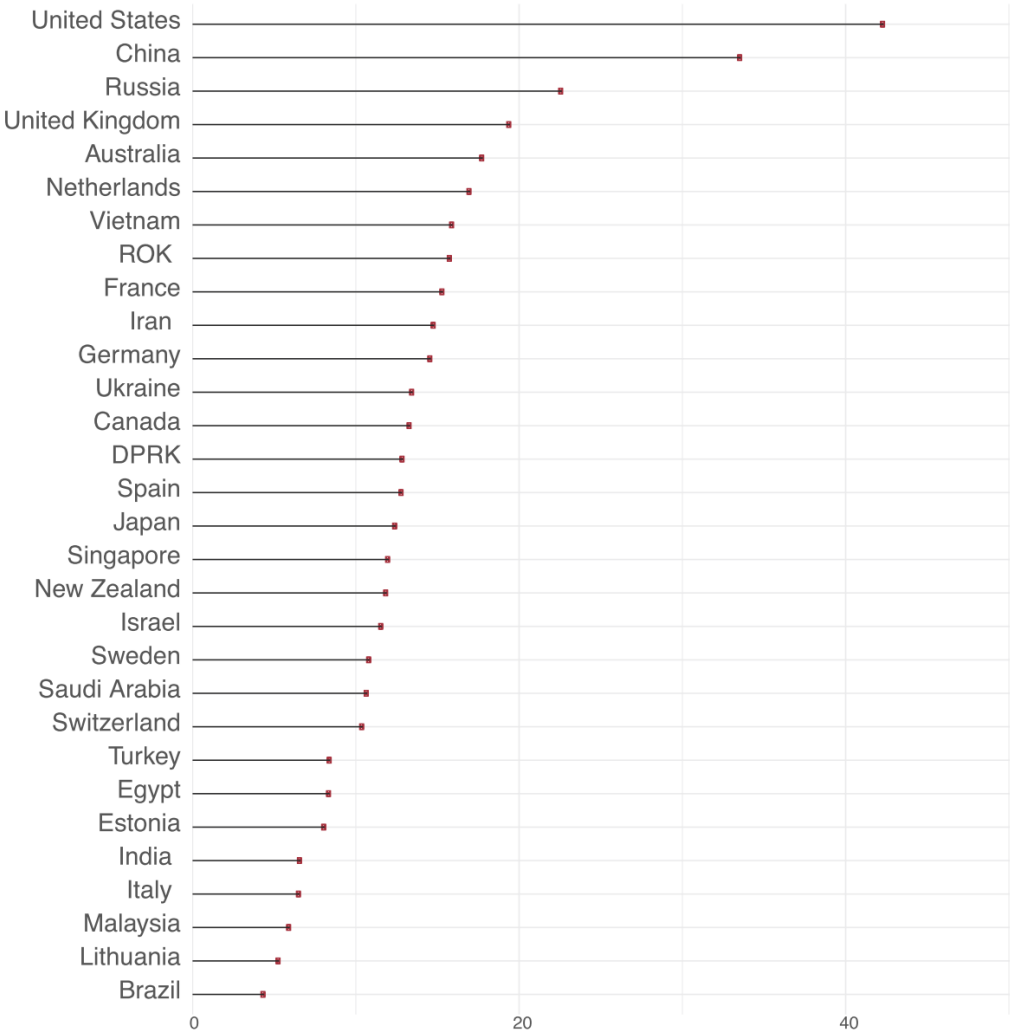
Critério 6 - Destruir ou desabilitar a infraestrutura de um adversário e suas capacidades: O país usou técnicas, táticas e procedimentos cibernéticos destrutivos para deter, corroer ou degradar a capacidade de um adversário lutar em domínios cibernéticos ou convencionais;

Critério 7 - Definir normas cibernéticas internacionais e padrões técnicos: O país participou ativamente de debates jurídicos, políticos e técnicos internacionais sobre normas cibernéticas.

Critério 8 - Acumular Riqueza e/ou Extrair Criptomoeda: O país conduziu operações cibernéticas para acumular riqueza.

Esses critérios possibilitaram a elaboração do índice de classificação geral do poder cibernético de 30 países conforme Figura 6.

Figura 6 – Classificação geral dos países



Fonte: [20]